International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

Autonomous Navigation in Dynamic Environment

K V Niranjan Gupta[1], K Prascheth[2], Vijaya Lakshmi K R[3], Sneha S Kadam[4], Dr Kiran Bailey[5]

[1]Student, Dept. Of ECE, BMS College of Engineering, Karnataka, India
[2]Student, Dept. Of ECE, BMS College of Engineering, Karnataka, India
[3]Student, Dept. Of ECE, BMS College of Engineering, Karnataka, India
[4]Student, Dept. Of ECE, BMS College of Engineering, Karnataka, India
[5]Assistant Professor, Dept. Of ECE, BMS College of Engineering, Karnataka, India

---

Abstract - *Robots have become an important part of our lives in recent years. They can be used for different purposes in different fields to make human life easier. Depending on the needs of the situation, robots can be used for complicated tasks to dangerous ones. Today we can see many jobs getting automated. Robotics has led to major automation in production and warehouse facilities. It is being adopted now into hospitals and surgeries too. Robots can achieve perfection levels which a normal human cannot. They also have other advantages like they can work for longer time and are cost effective in long run. These factors have led to the widespread use of robotics in the automation field. For any robot, navigating is an important task. The aim of this paper is to compare different combinations of path planning algorithms for dynamic environments. A dynamic environment may include a warehouse with people walking around or a cafeteria. Here the maps and scenarios keep changing and the robot must adapt to those changes so that it does not collide when an obstacle appears suddenly in front of it. We have implemented the path planning for a robot in a dynamic hospital environment. The algorithms are simulated using robotic simulation software like ROS, Gazebo and Rviz.*

Key Words: Robotics, Autonomous navigation, Path-planning, Dynamic environment, ROS

1. INTRODUCTION

Autonomous robots, such as industrial robots in factories and service robots in public areas, have attracted much attention and increasingly developed in the past few decades. These autonomous robots can navigate in locations dangerous to workers with minimum human intervention. Obstacle avoidance by mobile robot is of utmost importance as it ensures the safety of the humans and things in the surrounding and the robot itself, hence it becomes very important to choose the right algorithm. The ability to handle concave obstacles along with 'go to goal' behaviour is essential [1].

ROS based Autonomous Mobile Robot Navigation uses 2D LIDAR and RGB-D Camera for distance and depth, SLAM for Localization and Mapping. Adaptive Monte Carlo Localization can also be implemented with SLAM for localization of the mobile robot [2, 3]. RQT nodal graphs are used to demonstrate the interactions and visualizers like Rviz for visualization of the navigation. Autonomous Navigation of mobile robots is also possible in an uneven and unstructured environment [4]. This uses the concept of a multi-layered map to face an uneven environment which results in 3D representation of the environment in the real world. For better avoidance of obstacles, predictive and reactive based planning algorithms can be implemented. Model and learning based (reinforcement and supervised learning) techniques for navigation are used [5]. Implementation of different algorithms have their pros and cons.

In addition to remote environments, robots co-exist with humans in the same environment like that of a warehouse [6] and office [7]. In this system, the ARCO robot navigates in an environment that has obstacles and humans around. Lazy Theta* algorithm is being used as both global and local planner to achieve lower computation in order to ensure safety. Human-robot collaboration increases productivity and performance, and at the same time can cause hazardous situations to occur, which must be avoided. Risk assessment of such hazards is essential. Usage of scene graphs based on V-REP simulator is useful in the risk management process [8].

However, the implementation of autonomous navigation in a hospital environment is challenging [9]. The system is required to have as less computational and navigational time as possible to perform the assigned tasks. Hence, a highly efficient robust system becomes crucial.

This paper is organized into sections. Section 2 of the paper describes the methodology and implementation of the proposed solution. Section 3 discusses the results of the real time simulation. Conclusions and future work are presented in section 4.

2. METHODOLOGY AND IMPLEMENTATION

2.1 Problem Statement

Path planning between two points is a well-known NP-hardness [10]. As the degrees of freedom increases the complexity increases. While considering path planning in most of the scenarios the environment is restricted to 2-dimensional frame (X-Y axis) and Z is neglected. Due to this the 3 D obstacle gets missed out by the 2 D obstacle

detector like 2D Lidar [11] whereas 3D lidars are costly for all applications. During testing of the planning, the response of the system to dynamic obstacles of varied shape is not considered. Comparison of the local path planning algorithms along with global planners is important [12].

2.2 Proposed Solution

We propose a solution incorporating a PI based differential drive robot [13] that uses an RGBD camera attached at an elevation and 2D lidar sensor at the base of the mobile robot to capture the information of the environment. In the robot model the camera is given 1 degree of freedom for adjusting itself for the environment. The Lidar is used to get the 2 D information of the environment and RGBD is used to extract more information of the environment (Depth of obstacles in the surrounding with an elevation) that is used for path planning. This information of obstacles that are static as well as dynamic, helps in navigating autonomously without colliding the obstacles. We have used both IMU and encoder reading for the localization of the mobile robot. This setup is testing using open-source robotic software-ROS along with gazebo for the environment and Rviz for visualization.

The URDF (Universal Robot Description Format) is used to describe the physical model of a robot. The robot modelling can be done in SolidWorks and converted into this format for to be used ROS. It consists of description of a robot by its dimensions, type of joints between link and color of parts. This is fed to the robot state publisher and RVIZ. Using the robot state publisher package, we can publish the state of the robot to transform. Based on the URDF model, this package will publish the 3D pose of each link. This package also subscribes to joint states of the robot. RVIZ is the 3D visualization tool which we use to view the robot model, to read the sensor information and replay the data that was captured.

For sensing the obstacles, we use Lidar. The lidar scans the environment around the robot and also does the task of creating depth to laser scan. Depth is measured on scanning a 3D environment. An IMU sensor is an inertial measurement unit sensor which contains gyroscope, accelerometers and magnetometers to determine the orientation of the robot. Inertial Measurement Unit measures the velocity and the orientation of the module. By using the data from IMU and encoders, we can estimate the change in position over a period of time using Odometry. We use Odometry to estimate the position of the moving robot relative to the starting position. Rapid and accurate data and proper calibration is required for Odometry to work properly and give proper position estimates.

SLAM (Simultaneous localization and mapping) is used for localization and mapping purposes. SLAM takes inputs from the sensors and the transforms from the robot publisher. It will estimate the position of the robot with respect to the entire environment. This is required as we need to know the starting point for navigation. After localizations and mapping process is completed, the data is fed to the path planning algorithm.

Path planner gets information from SLAM and by using the starting position and destination, it creates a path to traverse by avoiding collisions. Path planner uses two plans. A global plan and a local plan. The planner calculates the values to be fed to the motors for the movement of the robot through a ROS serial to the microcontroller. The microcontroller then communicates with the actuators through a driver with the velocity command. This causes movement of the robot which can be visualized in RVIZ. The actuators' velocity, acceleration and angular velocity is gathered by encoders and given as a feedback to the system. Figure 1 shows the block diagram of the proposed system.

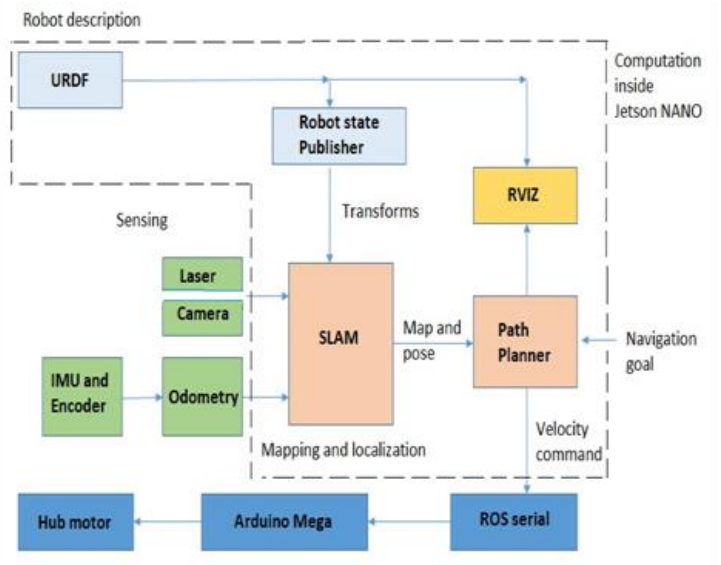

Fig -1: Block Diagram of the robot

The software architecture of the robot is a layered architecture consisting of four main layers – Application layer, System service layer, OS layer and Hardware abstraction layer. Figure 2 shows the four layers.

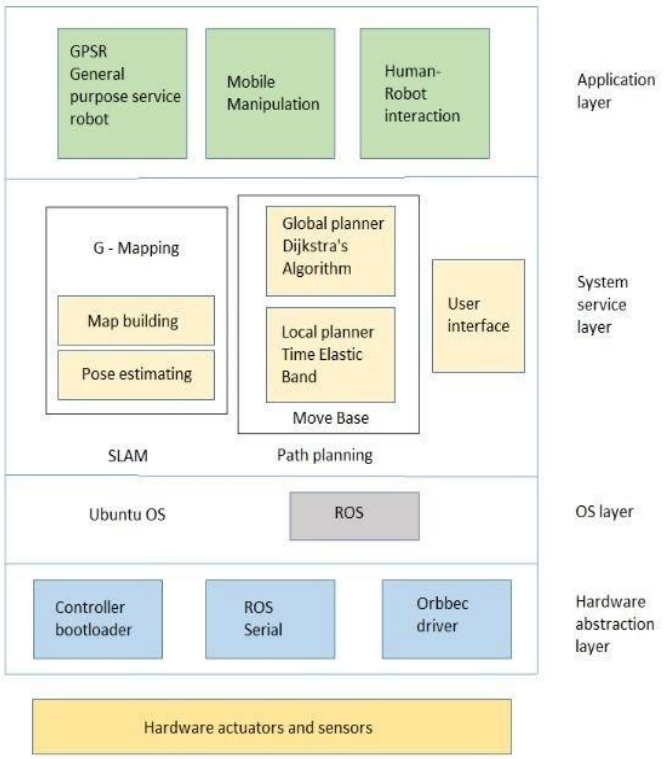

Fig -2: Software Architecture of the robot

2.3 Algorithms

The different algorithms implemented are described in this section.

1) Local planners

a) TEB: TEB (Time Elastic Band) is a local planner which creates a sequence of intermediate robot poses by changing the global plan created initially [14]. Velocity limits, acceleration limits, security distance from obstacle, kinematic, geometric, and dynamic constraints of vehicles form a part of the cost function. These inputs to the planner will generate a set of velocity commands and steering angles which are needed to achieve intermediate waypoints as the robot moves. Three elastic bands are created- one updated in each iteration, second is calculated from the original planned path and third is created from scratch. The shortest one of these three is selected so that it satisfies all the requirements.

b) DWA: DWA (Dynamic Window Approach) performs sample-based optimization. It simulates paths in the feasible velocity space according to the horizon lengths based on robot's motion model. It cannot predict motion reversals as the control action is kept constant. This approach gives decent performance for differential drive and omnidirectional robots. It is well suited for grid-based evaluations. DWA does not get stuck in a local minimum unlike TEB algorithm. Suboptimal solutions based without motion reversals and support for non-smooth cost functions are main features of DWA.

c) Eband: EBAND (Elastic Band) searches for the path while expanding to both sides with force. This behavior is similar to a rubber band. Eband generates a path with a pulling force thereby reducing the search path. When more obstacles are encountered, path is changed using same principle. An elastic band is computed within the local cost map. Robot tries to follow the path by connecting the center points of the band.

2) Global planners

a) Dijkstra's algorithm: Dijkstra's algorithm is one of the simplest global planners. The algorithm starts from the starting point of the route and marks all immediate neighbors of the initial vertex with the cost to get there. It then moves from the vertex with the least cost to all of its adjacent vertices and assigns them the cost to get to them via itself if this cost is lower. The algorithm proceeds to the vertex with the next lowest cost after all neighbors of a vertex have been checked. Once the goal vertex is reached the algorithm terminates and the robot can follow the edges pointing towards the lowest edge cost.

b) A* algorithm: A* Search algorithm is one of the popular techniques used in graph traversals. At each step the algorithm selects the node according to a value - 'f' which is "g+h". At each step it selects the cell having the lowest 'f'. We define 'g' and 'h' as below g = the cost to traverse from the initial point to a given square on the grid, by following the path generated to get there. h = the predicted cost to move from that given square on the grid to the destination. This is often called as heuristic.

c) Navfn: A quick interpolated navigation function is provided by Navfn, which can be used to make plans for a mobile base. The planner will assume a circular robot and operate on a cost map. It creates a minimum cost plan from a start point to an end point in a grid. The navigation function is computed with Dijkstra's algorithm.

Fig -3: Environment of Gazebo

3. RESULTS

As the planner does not give same result in a dynamic environment, the results tabulated below are for one of the iterations. The robot traverses to different locations and reaches the goal in the environment. Walking human models in the world represents the dynamic object in the environment.

The table 1 tabulates the global and local planners used, total distance traversed, time taken and efficiency of the planner. The efficiency of planner combinations is calculated by normalizing the product of time and distance with the maximum product among the set of algorithms.

We can infer that a good combination of planners in terms of efficiency in both distance and time among the set of algorithms in dynamic environment is 'TEB' as local planner and 'A star' as global planner. To tackle the dynamic obstacles the local planner plays a very crucial role. A higher priority should be given for the selection of local planners.

Table-1: Comparison of different Planning Algorithms in Dynamic Environment

Planners		Results		
Local	***Global***	***Distance***	***Time***	***Efficiency***
DWA	Navfn	35.5	238	0.296
Eband	Navfn	37.34	221	0.303
TEB	Global Planner (Dijkstra)	35.74	106	0.661
TEB	Global Planner (A star)	27.26	67.2	1

During the implementation, despite extensive tuning, it was observed that DWA planner was not suitable for dynamic environment, as it constantly gets stuck and recovery behavior is initiated which consumes higher time to reach the goal points. In Eband planner the time of execution increases as the robot rotates when it approaches the goal positions when the lateral tolerance is within the range. The A star algorithm in global planner reduces the number of paths to be computed in planning when compared to Dijkstra's algorithm and hence requires less time. Hence, we can observe higher efficiently for this pair when compared to other pairs.

4. CONCLUSIONS AND FUTURE WORK

We can conclude that the autonomous navigation is implemented with the map build used in SLAM algorithm (gmapping) and different algorithms are tested in the dynamic hospital environment. In a dynamic environment the local planner plays a very important role. The evaluation of the algorithm has to be done by considering both optimal path with shortest time for the mobile robots. In light of study findings, it is observed that TEB along with Global planner (A Star) is a good combination for implementation of robot in dynamic environment which has high efficiency in terms of both time and distance. This can be used for a quick comparison criterion before implementation.

The present work can be carried out in hospitals, where mobile robots can be assigned as nurses during tough times like that of a pandemic. This can be enhanced to implement a sound-based localization and goal position identification which identifies the source of sound and navigates to it. Task based navigation can be implemented for the mobile manipulators.

REFERENCES

[1] S. Khan and M. K. Ahmmed, "Where am I? Autonomous navigation system of a mobile robot in an unknown environment," 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 2016, pp. 56-61, doi: 10.1109/ICIEV.2016.7760188.

[2] S. Gatesichapakorn, J. Takamatsu and M. Ruchanurucks, "ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 2019, pp. 151-154, doi:10.1109/ICA-SYMP.2019.8645984.

[3] Thale, Sumegh & Prabhu, Mihir & Thakur, Pranjali & Kadam, Pratik. (2020). ROS based SLAM implementation for Autonomous navigation using Turtlebot. ITM Web of Conferences. 32. 01011. 10.1051/itmconf/20203201011.

[4] C. Wang et al., "Autonomous mobile robot navigation in uneven and unstructured indoor environments," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 109-116, doi: 10.1109/IROS.2017.8202145.

[5] J. Cheng, H. Cheng, M. Q. -Meng and H. Zhang, "Autonomous Navigation by Mobile Robots in Human Environments: A Survey," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 1981-1986, doi: 10.1109/ROBIO.2018.8665075.

[6] Rey, Rafael & Corzetto, Marco & Cobano, J. & Merino, Luis & Caballero, Fernando. (2019). Human-robot co-working system for warehouse automation. 578-585. 10.1109/ETFA.2019.8869178.

[7] P. Alves, H. Costelha and C. Neves, "Localization and navigation of a mobile robot in an office-like environment," 2013 13th International Conference on Autonomous Robot Systems, 2013, pp. 1-6, doi: 10.1109/Robotica.2013.6623536.

[8] Inam, Rafia & Raizer, Klaus & Hata, Alberto & Souza, Ricardo & Forsman, Elena & Cao, Enyu & Wang, Shaolei. (2018). Risk Assessment for Human-Robot Collaboration in an automated warehouse scenario. 743-751. 10.1109/ETFA.2018.8502466.

[9] M. B. Alatise and G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," in IEEE Access, vol. 8, pp. 39830-39846, 2020, doi: 10.1109/ACCESS.2020.2975643.

[10] Erickson, L.H. & Valle, S.M.. (2013). A simple, but NP-hard, motion planning problem. Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013. 1388-1393.

[11] C. Zhang, J. Wang, J. Li and M. Yan, "2D Map Building and Path Planning Based on LiDAR," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 783-787, doi: 10.1109/ICISCE.2017.167.

[12] M. Korkmaz and A. Durdu, "Comparison of optimal path planning algorithms," 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), 2018, pp. 255-258, doi: 10.1109/TCSET.2018.8336197.

[13] S. Gupta, "Autonomous path planning of differential drive robots for coordinate based navigation," 2016 IEEE International Conference on Mechatronics and Automation, 2016, pp. 563-568, doi: 10.1109/ICMA.2016.7558625.

[14] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T. Bertram, "Efficient trajectory optimization using a sparse model," 2013 European Conference on Mobile Robots, 2013, pp. 138-143, doi: 10.1109/ECMR.2013.6698833.