International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-12 Issue-2, July 2023

Autonomous Robot Navigation in Known Environment

Fazina Kosser, Neerendra Kumar

Abstract: ***Autonomous robot navigation is one of the challenging researched topic in robotics. A secure and optimal path is required for any mobile robot navigation in a known environment. In this work, a Simulink model is proposed based on Pure Pursuit and path-following controllers for solving the navigation problem of mobile robots in a known environment. The Pure Pursuit controller is used to determine the linear and angular velocities of the robot. Moreover, the (x, y) coordinate position of the robot and the waypoints are input to the pure pursuit block. Velocity commands are sent to drive the robot on the given path. The primary objective of the proposed model is to determine an obstacle-free path for mobile robot navigation. However, the robot must navigate from the start to the target location without hitting obstacles. For experimental results, the Turtle Bot Gazebo simulator is used. The "Robotic System Toolbox" in MATLAB is used to program the navigation process.***

Keywords: Autonomous, Localisation, Motion Control, Navigation, Path Planning and Search Algorithms.

I. INTRODUCTION

Nowadays, robots play a vital role in our society. This is because humans are being replaced by robots in a wide range of activities, from basic to highly complex and dangerous ones. Navigation is one of the most interesting researched topics in autonomous robotics. Navigation of robots is a common problem in these applications. Primarily, localisation and path planning must be incorporated into the robot's navigation. Grid map-based navigation is widely known as one of the effective ways [1] [2]. The grid maps are typically constructed using SLAM and PRM algorithms. However, creating occupancy grid maps in the real world is a challenging task. Occupancy grid map-based navigation can provide precise localisation and path planning. However, grid map-based navigation has several problems. Creating precise grid maps of large environments is a time-consuming process. Furthermore, velocity commands are not directly sent to the robot. The navigation task is completed using a grid map, which is less time-efficient. A robot system that does not depend on such precise grid maps is desired. This study aims to propose a navigation system that does not rely on grid maps. Furthermore, if obstacles are present in the path, there is a chance that the robot will collide with them.

Manuscript received on 28 February 2023 | Revised Manuscript received on 11 July 2023 | Manuscript Accepted on 15 July 2023 | Manuscript published on 30 July 2023.
*Correspondence Author(s)
Fazina Kosser*, Department of Computer Science and Engineering, IT from Central University of Jammu, India. E-mail: fazinakosser157@gmail.com, ORCID ID: 0000-0003-3006-1050
Neerendra Kumar, Department of Computer Science and Engineering, IT from Central University of Jammu, India.

Navigation aims to find the secure and optimal path for a mobile robot. The navigation process is primarily based on the following steps: creating a map of the environment, saving that map in the corridor world, identifying a feasible path from the start to the target within the given map, and finally, driving the robot along that feasible path. Driving a robot along a given path without colliding with obstacles in a static environment is a challenging task. In [3], a bio-inspired algorithm based on differential evolution is presented. The job is to check the next free position on each iteration and minimise the objective function based on Euclidean distances. Petri nets are used for obstacle avoidance as presented in [4]. The supervised system is used for obstacle avoidance in a known environment. Path planning is a crucial parameter for determining the optimal path between the source and destination. An optimal path may be defined as the path that minimises translation and rotation. Currently, path planning is the most researched topic in mobile robot navigation [5]. To solve the problem of path planning, an algorithm based on free segments and turning points in a known environment was presented in [6]. This algorithm aims to find a feasible path between the source and the goal. It also reduces path length while avoiding collisions with obstacles. An evolutionary algorithm is used for robot navigation, which makes use of sensors and motors to control the robot, as stated in [7]. Navigation based on odometry and global positioning is presented [8]. Wireless sensors are used for navigation in a collision-free path. In [9] various algorithms based on path planning techniques are compared and verified. After analysis, the JPS algorithm is used to find the shortest path quickly in a static environment only. Moreover, where there is no real-time analysis and the only objective is to find the shortest route, then Basic theta* is recommended. The HCTNav algorithm for robot navigation is a cost-effective technique for indoor environments. It also shows better results than Dijkstra's algorithm, as stated in [10]. A heuristic approach and map building are presented in [11].

II. PRELAM INARY WORK

A. Environment Setup

To set up the environment, we first bring up a robot in the environment. Then, we take a Simulated Turtlebot in the Corridor world of the Gazebo Simulator. The command must be executed in the Ubuntu terminal, as shown below. As an output, the corridor world of the Gazebo simulator with a simulated Turtlebot opens and is given as follows in Fig. 1.

Command to bring up 'Corridor World' Roslaunch turtlebot_gazebo turtlebot_world.launch (File name)

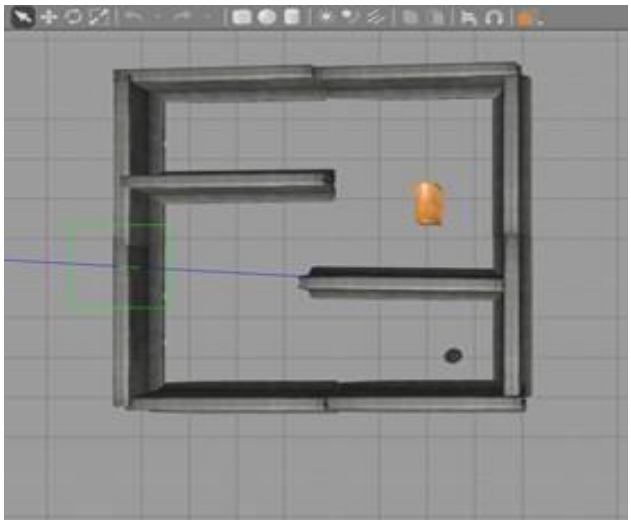

Fig. 1. Command to bring up 'Corridor World'

B. Optimal Waypoints Extraction

Once the desired Corridor World of Gazebo Simulator is created, there is a need to capture the (x, y) coordinates of the environment by moving Turtlebot using the Keyboard Teleoperator command. Now, we need to extract optimal waypoints by moving the TurtleBot using the keyboard teleoperator command presented. By executing the following keyboard-teleoperator command:

Roslaunch turtlebot_teleop keyboard_teleop.launch

The operator command we perform enables us to control the robot and move it within the corridor world. For waypoint extraction, a MATLAB code is executed to write the desired (x, y) coordinate position of the robot to a text file. The data stored in the text file is used to provide waypoints of the desired path that the robot can follow while moving from the source to the goal. Waypoints used in this model start position (-0.0000 -0.0000) and goal position (3.9341, 5.0792).

III. ALGORITHM FOR PROPOSED WORK

The following steps are to be followed by the robot for path planning and navigation from the start to the goal location.

1) Determine the current position of the robot in the actual environment.
2) Compute the linear and angular velocities of the robot.
3) Send these velocity commands to the robot.
4) Check whether the robot is close to the goal stop; if not, then
5) Repeat the process until the robot reaches the goal location.

IV. PROPOSED WORK

A. Proposed Simulink model

Figure 2 shows the proposed Simulink model. The proposed model implements a path-following controller based on the pure pursuit algorithm. The controller receives the robot's pose and scans data from the simulated robot, then sends velocity commands to drive the robot along the given path.

The subscriber receives messages sent on the "/Odom" topic. The odometry data of the robot is sent on the topic "/Odom". The (x, y) location of the robot is then extracted from Msg of "/Odom". A new message is sent to collect new sensor information. Quat2Eul is a MATLAB function which converts the Quaternion function to Euler angles. The yaw orientation of the robot is then extracted from the pose message. The path is specified as a set of waypoints that the robot follows. Waypoints are considered to be two-dimensional coordinate positions of the robot. In the given environment, waypoints are considered at different points. Among various points, one is the robot's starting position. The second waypoint is the robot's goal or target location. The other one is placed near the obstacles to determine whether the robot avoids them or not. Pure Pursuit receives two inputs: one is the (x, y) position of the robot and the set of waypoints. The Pure Pursuit block computes the linear and angular velocities of the robot. Moreover, without considering obstacles, the robot can navigate from the source to the goal location. A path-following controller is used in this model to follow the waypoints, allowing the robot to navigate the given environment while avoiding obstacles along the path. To stop the robot once it reaches at goal location. The goal is the last waypoint on the route. This Pure pursuit also compares the current robot position and the goal point to determine if the robot is close to the goal. The 'Outputs' subsystem publishes the linear and angular velocities to drive the simulated robot. It adds the velocities computed using the Pure Pursuit algorithm. The final velocities are set on the "Odom" message and published on the topic. This is an enabled subsystem that is triggered when a new laser message is received. This means that a velocity command is published only when new sensor information becomes available. This prevents the robot from hitting obstacles in the event of a delay in receiving sensor information. Scope is used to plot the angular and linear velocities of the robot over time.

B. Path Pursuit

Once the feasible path from the start location to the target location in the given environment is determined, the robot simply follows the waypoints to reach the target location. Path-following controllers based on the pure pursuit algorithm are used to find a feasible path.

Retrieval Number: 100.1/ijrte.F75050311623
DOI: 10.35940/ijrte.F7505.0712223
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

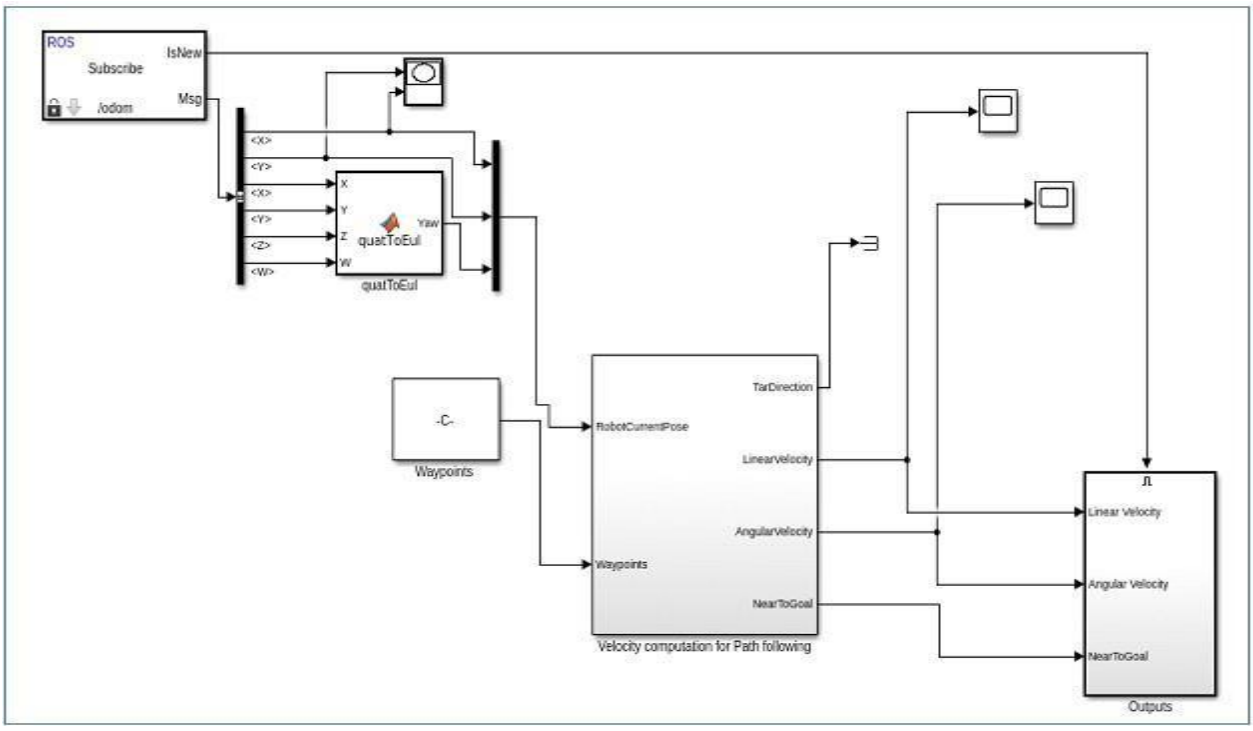

Fig. 2. Proposed Simulink Model

V. RESULTS AND DISCUSSION

In Fig1. Shows the corridor world when the robot is at the start location. Upon executing the proposed Simulink model, the robot navigates successfully by following the set of waypoints and reaches the goal. In Fig. 3, it indicates that the robot has reached the goal location. The robot navigates successfully by avoiding obstacles along its path. The path followed by the robot is being plotted, as presented in Fig. 4. The linear and angular velocities with respect to time are also plotted as presented in Fig. 5 and Fig. 6, respectively. The robot uses these velocities to follow the waypoints and reach the destination; the angular velocity changes at each time interval. Moreover, linear velocity also shows minor fluctuation with respect to time but remains constant throughout the navigation process.

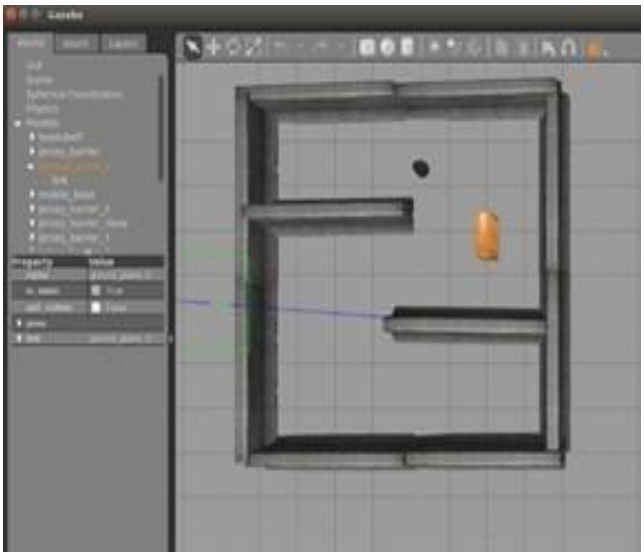

Fig. 3. 'Corridor World' when the robot is at the goal location

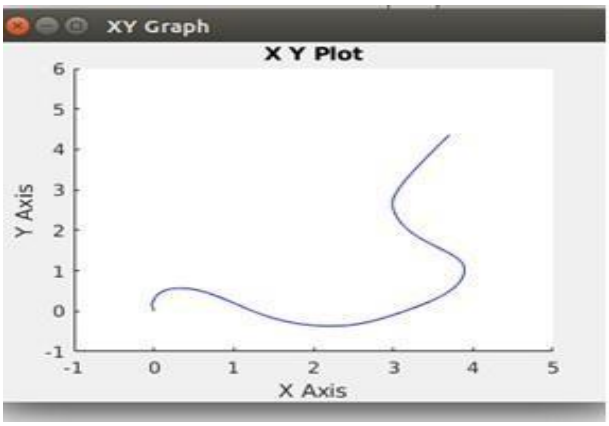

Fig. 4. Path plot followed by the Robot from source to goal

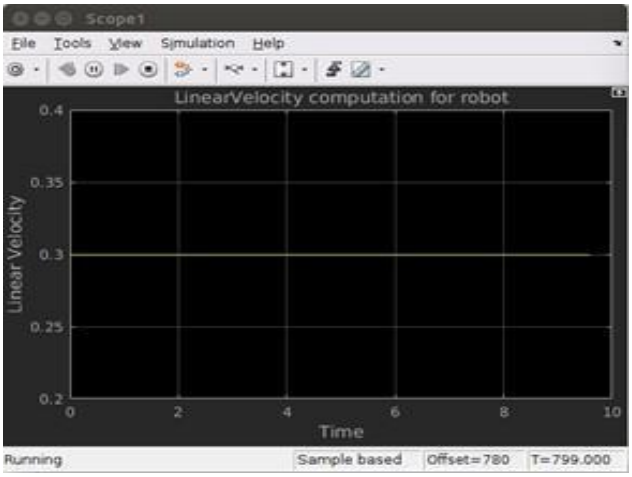

Fig. 5. Linear Velocity computation for Robot

Retrieval Number: 100.1/ijrte.F75050311623
DOI: 10.35940/ijrte.F7505.0712223
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

VI. EXPERIMENTAL SET-UP

The experimental work is carried out on a computer equipped with a dual-core 2.4 GHz Intel i7 Processor and 4 GB of RAM. The operating system used here is Ubuntu 16.04 LTS. The version of Robot Operating System (ROS) is Kinetic (1.12.2). The TurtleBot-Gazebo simulator, version 7.0.0, is considered for the implementation of the proposed model. MATLAB R2018 is used for programming purposes. The "Robotic System Toolbox" in MATLAB is used to program the navigation process.

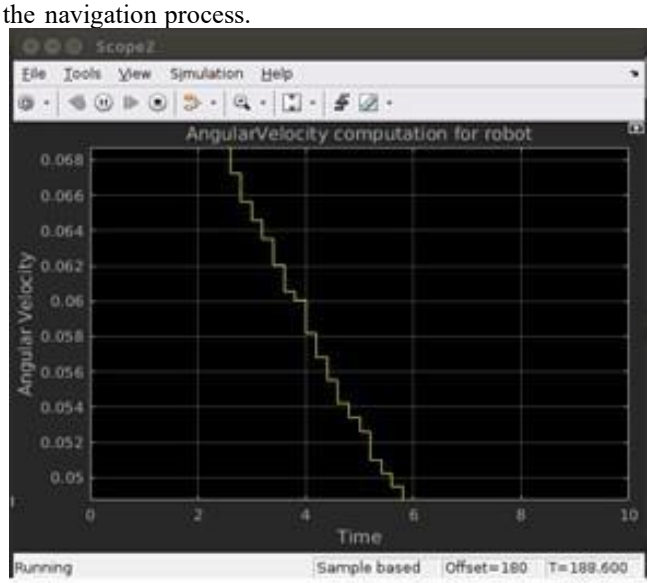

Fig. 6. Angular Velocity computation for Robot

VII. CONCLUSION AND FUTURE SCOPE

The proposed model can avoid the obstacles present in the path. This model operates in a known environment and follows the waypoints. Moreover, the robot navigates successfully from source to goal and avoids obstacles along the route, regardless of their shape and size. When compared with the grid map-based technique, the proposed method is time-efficient. In a Known environment map, the environment is predefined. This model works in a known environment, where prior information is available. In the future, we plan to extend this work to a dynamic environment where little to no previous information is available. Simulation results are presented using TurtleBot Gazebo to demonstrate that the proposed method is a viable alternative for solving the robot navigation problem in a known environment.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	All authors have equal contributions to this article.

REFERENCES

1. N. Kumar, Z. Vamossy, and Z. M. S. Resch, "Robot path pursuit using probabilistic roadmap," in 2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI), Nov 2016, pp.000 139–000 144. [CrossRef]
2. N. S. N. Buniyamin, W.A. J. Wan Ngah and Z. Mohamad, "A simple local path planning algorithm for autonomous mobile robots," INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT, vol. 2, pp. 151–159, 05 2011.
3. E. G. Martinez-Soltero and J. Hernandez-Barragan, "Robot navigation based on differential evolution," International Federation of Automatic Control, vol. 51, no. 13, pp. 350 – 354, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896318310565 [CrossRef]
4. O. F. S. Bartkevicius, A. Knys, A. Derviniene, G. Dervinis, V. Raudonis, A. Lipnickas, V. Baranauskas, K. Sarkauskas, and L. Balasevicius, "Mobile robots navigation modelling in known 2d environment based on Petri nets," Intelligent Automation and Soft Computing, pp. 1–7, 2017. [Online]. Available: https://doi.org/10.1080/10798587.2016.1264695 [CrossRef]
5. F. Kosser and N. Kumar, "Robot navigation and path planning techniques challenges: A review," International Journal of Electronics Engineering, vol. 11, pp. 115–125, 06 2019. [Online]. Available: http://www.csjournals.com/?p=2816
6. I. M. Imen Hassani and C. Rekik, "Robot path planning with avoiding obstacles in a known environment using free segments and turning points algorithm," Mathematical Problems in Engineering, no. 13, 2018. [CrossRef]
7. C. J. C. Lucas da Silva Assis, Anderson da Silva Soares and J. V. Baalen, "An evolutionary algorithm for autonomous robot navigation," Procedia Computer Science, vol. 80, pp. 2261 – 2265, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916308808 [CrossRef]
8. J. Freeman and S. Surendran, "Remote monitoring of indoor environment using mobile robot-based wireless sensor network," Advances in Robot Navigation, 08 2011. [CrossRef]
9. Duchon, H. Frantisek, D. Dominik, B. Martin, and Andrej, "Optimal navigation for mobile robot in known environment," Applied Mechanics and Materials, vol. 282, pp. 33–38, 01 2013. [CrossRef]
10. N. O. E. F. L.-C. A. S. A. d. C. Marco Pala and J. Garrido, "Hctnav: A path planning algorithm for low-cost autonomous robot navigation in indoor environments," ISPRS International Journal of Geo-Information, pp. 729–748, 8 2013. [CrossRef]
11. N. Kumar, Z. Vámossy, and Z. M. Szabó-Resch, "Heuristic approaches in robot navigation," in 2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES), June 2016, pp. 219–222. [CrossRef]

AUTHORS PROFILE

Fazina Kosser has completed M.Tech in Computer Science and Technology from the Department of Computer Science and IT at Central University of Jammu, India. She has completed a B.Tech in Computer Science and Engineering from Baba Ghulam Shah Badshah University, Rajouri, India. Her research interests include robotics, Data mining, and Artificial Intelligence.

Neerendra Kumar is working as an Assistant Professor at the Computer Science and IT department of Central University of Jammu, India. He completed his PhD studies at Óbuda University, Budapest, Hungary, under the Stipendium Hungaricum Scholarship. His teaching interests include programming languages, Algorithms, Artificial Intelligence, and Robotics. His current research area is Robot Navigation and Path Planning. The details of his faculty profile are available at http://cujammu.ac.in//5084/5084_media/nareendra.pd.f

Retrieval Number: 100.1/ijrte.F75050311623
DOI: 10.35940/ijrte.F7505.0712223
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Retrieval Number: 100.1/ijrte.F75050311623
DOI: 10.35940/ijrte.F7505.0712223
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)