DESIGN AND DEVELOPMENT OF AUTONOMOUS UNMANNED GROUND VEHICLE USING PX4

Project Reference No.:*45S_BE_0044*

College ***:*** *Sahyadri College of Engineering & Management*
Branch ***:*** *Department of Electronics and Communication Engineering*
Guide(s) ***:*** *Mrs. Megha N*
Student(S) ***:*** *Mr. V Mukesh Kumar*
Mr. Jeevan K
Mr. V H Darshan

Keywords

Autonomous, Robotics, Navigation, Industrial Automation

Introduction

Nowadays, Unmanned Ground Vehicles are widely used for several purposes. Collaboration on various kinds of robots from different makers or even self-developed robots in the same ecosystem are trending. Manufacturing and Industrial Robotics have reached a point where to be more useful to small and medium sized manufacturers, the systems must become more agile and must be able to adapt to changes in the environment. In this context, we proposed a solution of developing an autonomous robotic platform capable of doing multiple tasks thus avoiding unnecessary repetitive work, contact between humans and helps to increase productivity. Compared with traditional human work, robots and autonomous systems have advantages such as the ability to perform tasks 24x7 with high accuracy, efficient path planning, increased flexibility and safety. We have designed an autonomous robotic platform consisting of a differential drive mobile base equipped with intelligent sensors aiding in performing the operation.

Objective

(a) Designing and developing an autonomous robotic platform consisting of a differential drive mobile base using Robot Operating System which can be used to undertake repetitive tasks

(b) Autonomous navigation throughout the operational area, where a common mobile base along with many specific extensions is used to perform different operations.

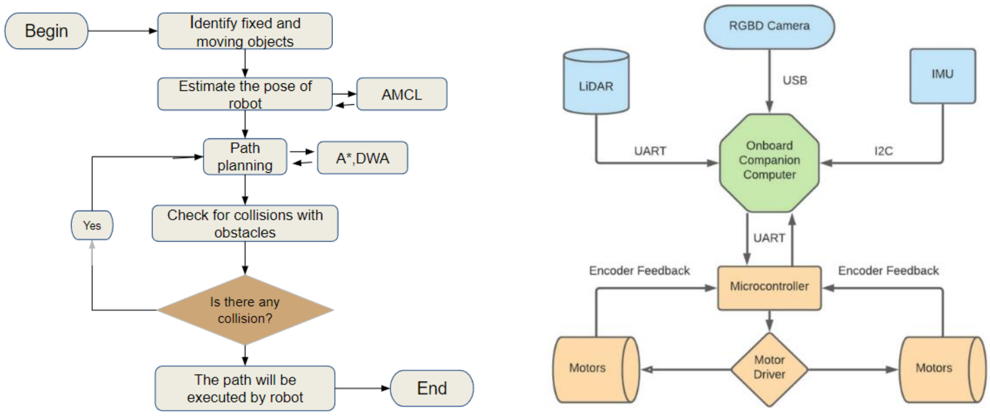
(c) The autonomous robotic platform consists of a differential drive mobile base equipped with intelligent sensors aiding in performing the operation.

(d) Fuse data from different sensors to get accurate odometry and thus leading to increased robustness

Methodology

The robot performs complete autonomous navigation throughout the operational area, where a common mobile base along with many specific extensions is used to perform different operations. The navigation problem mainly deals with two things: localization and path planning. Our approach involves SLAM (Simultaneous Localization and Mapping) to get the initial map of the surrounding environment. SLAM is a computational problem that involves creating or updating a map of an unknown environment while also tracking the position of an agent within it. While this appears to be a chicken-and-egg problem at first glance, there are several algorithms known for solving it in tractable time in certain environments. The SLAM algorithm simultaneously maps the environment and localizes the robot by utilizing laser scan data from the LiDAR sensor and odometry data from the wheel encoders. The initial map is later used for path planning to move from the initial pose to the goal location. AMCL (Adaptive Monte Carlo Localization) algorithm is used to get the pose of a robot in a given map. AMCL works with the help of particle filtering. In particle filtering, particles are scattered randomly in the environment. During each periodic iteration, the particles obtain a higher amount of weight whenever it estimates the pose of the laser. The particle is reassembled according to the updated weights, while they are converging to a specific point, then that corresponding position is the maximum likelihood position of the robot. For path planning and static obstacle avoidance A star (A*) grid-based algorithm is implemented on the robot. A star search algorithm is one of the most effective and widely used techniques for path discovery and graph traversal. A* is an informed search algorithm, or a best-first search algorithm, in the context that it is described in terms of weighted graphs: it seeks the shortest path to a specified goal node starting from a specific starting node in the graph. This is accomplished by maintaining a tree of paths that begin at the beginning node and trying to extend them one edge at a time until the process has finished. As a result, the shortest distance between start and objective is covered, as well as faster real-time execution.

We employ DWA (Dynamic-Window Approach) running parallel to A star to avoid Dynamic Obstacles that occur suddenly, allowing the robot to avoid any type of dynamic obstacle. This increases the robustness of the system in real- world scenarios. If the robot encounters any barriers in its path while navigating in the unknown environment, it recalculates its course to its destination.



Block diagram of Software Architecture and Hardware architecture

Results and Conclusion

The robot was designed and imported into custom gazebo simulation environment. We have successfully completed real time simulation of our robot in gazebo simulator. The robot is capable of performing point to point navigation and can localize itself using Robot Operating System wrappers. Dynamic obstacle avoidance is tested in different simulation environments. we have successfully completed real time simulation of an autonomous unmanned vehicle in Gazebo simulator. The vehicle is capable of performing point to point navigation and self-localization using Robot Operating System wrappers. Dynamic obstacle avoidance is tested in different simulation environments. The accuracy was validated by plotting odometry and estimated pose values. Real-world prototype of the robot was developed and tested for different scenarios with fixed and moving obstacles. Destination point was given as input in RViz and the robot computes shortest path from initial point to destination point. If the destination is unreachable, then the robot would traverse to the nearest free space.

Scope for Future Work

(a) Further in future we will be developing a prototype for testing in a real-world scenario and we will be using deep learning techniques for aiding in navigation tasks.

(b) Using reinforcement learning techniques for solving the navigation problem and comparing it with this approach

(c) Implementing the navigation stack using only camera, which would cut down the need of several other sensors.