Path Planning of Autonomous Drones using ROS

Kartik Hitesh Soni
Dept. of Mechanical Engineering
MIT School of Engineering
Pune, Maharashtra
Email: kartiksoni010@gmail.com

Anish Shivaji Wagh
Dept. of Mechanical Engineering
MIT School of Engineering
Pune, Maharashtra
Email: anish.wagh2002@gmail.com

Aaryan Sheetal Murgunde
Dept. of Mechanical Engineering
MIT School of Engineering
Pune, Maharashtra
Email: aaryan.murgunde@gmail.com

Nishad Milind Rajhans
Dept. of Mechanical Engineering
MIT School of Engineering
Pune, Maharashtra
Email: nishad07112000@gmail.com

Bhumeshwar Patle
Dept. of Mechanical Engineering
MIT School of Engineering
Pune, Maharashtra
Email:bhumeshwar.patle@mituniversiy.edu.in

Vishwananth Patil
Dept. of Mechanical Engineering
MIT School of Engineering
Pune, Maharashtra
Email:vishwanath.patil@mituniversity.edu.in

***Abstract*— The concept of UAVs (Unmanned Aerial Vehicles) has always been a hot topic of discussion among roboticists. The idea of engineering an aerial vehicle that can hover and maneuver around a certain region has a wide range of applications. However, the current drones in the market are made for a specific task orientation and are not very user customizable. The purpose of this research thus involves introducing the highly customizable interface of ROS into the drone automation firmware and come up with various optimized path planning algorithms for its navigation.**

Keywords—UAVs, drones, customizable, ROS, path-planning.

I. Introduction

A drone is a flying robot that may be remotely controlled or fly autonomously utilising software-controlled flight plans in its embedded systems, which work in concert with onboard sensors and a global positioning system (GPS). Drone development has always piqued engineers' interest due to its vast range of uses, which include mapping and surveying, delivery, agricultural use, aerial photogrammetry, and even military combat reactions. It certainly has been and will always be a prime topic of interest among the roboticists. One of the most important considerations when automating a machine is to provide it a path to follow. Getting a drone to fly high in the sky isn't difficult; nevertheless, getting a drone to make judgments on its own and navigate properly to the required places is what drone automation is all about. Path planning is an intriguing, albeit tough, task for any robot. Choosing a path planner that is properly optimized and efficient ensures a solid and safe navigation.

The current state of the autonomous drone market is difficult. Drones on the market are designed for specific tasks and are therefore difficult to customize. Drone planning systems are designed to navigate to a certain set of waypoints. They are not flexible enough to be able to use according to the user's requirement. This is where the introduction of ROS kicks in. It provides the flexibility of using multiple algorithms simultaneously and also enables a lot of other developing options. The purpose of this research paper is to examine all the challenges drone planning faces, and come up with their solutions while using ROS at the core of drone autonomy [1].

II. Path Planning using ROS

A. Introduction to Path Planning

Path planning, also called motion planning, is a computational problem that involves determining a set of feasible configurations to move an object between two locations. Finding a geometric path that connects the robot's current location to the goal based on a map is the purpose of a path planning algorithm. Furthermore, a prior knowledge of the environment is usually absent or partial in mobile robots operating in unstructured environments, or in service and companion robots, the environment is not static, i.e., during motion, the robot may encounter other robots, humans, or pets, and execution is frequently the result of uncertainty. Local obstacle handling, which includes obstacle detection and avoidance, is also required to achieve collision-free path planning. Robots may now detour around barriers utilising modern approaches by quantitative measurement of dimensions of obstacles.[2]

B. The ROS Framework

In order to simulate the drone, the proposed algorithms were implemented in the Robot Operation System (ROS), the open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. More importantly, ROS has plenty of opensource packages including sensor drivers, navigation tools, environment mapping tools, path planning tools, communication and visualisation tools and many others that ultimately rigidifies the drone software network[4].

Using Ardupilot, the unmanned vehicle autopilot software suite in the Gazebo environment[5], the drone receives its position from a MavROS node that connects to ROS for continuous movement along its x, y, and z axes. In a SITL simulation, the flying stack runs on the computer (either on the same computer or another computer on the same network). Sensor data is observed on the computer from the flight dynamics model in the flight simulator during SITL operations[4].

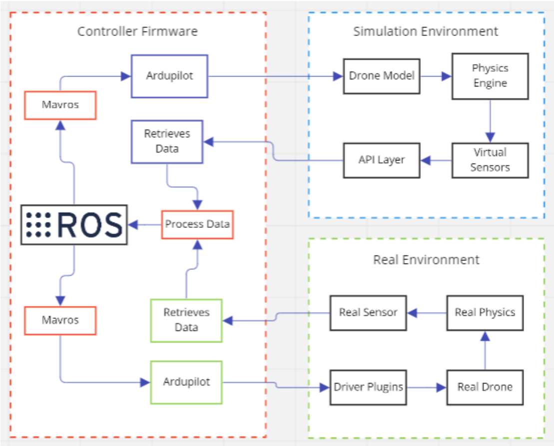


Fig 1 ROS-drone framework

C. Drone Construction

Mobile robots are required to know their locations within the environment as well as their surroundings so that they can perform assigned tasks. These issues are investigated within the context of localization and mapping, a phenomenon in robotics that analyses the world around a mobile robot. The method is implemented in software that runs on the ROS platform. By using a stereo depth sensor on the drone, the point cloud about the obstacles is obtained. For the simulation, the drone is equipped with various sensors for obstacle identification. The drone is equipped with 360 Lidar[11], it is a remote sensing technology where the environment is scanned with a pulsed laser beam and the reflection time of the signal from the object back to the detector is measured. Lidar sensors have obstacle detection capabilities over a wide field of view, which make them ideal as part of a sense and avoid solution. Additionally, ultrasonic sonic sensors and a depth camera are also used to determine the presence of obstacles and its data will be used to plan the optimal path. A number of filter operations are used to convert this data. The drone receives the user's intended destination information, which is used in conjunction with the drone's position and map information to construct the intended flight path[5].

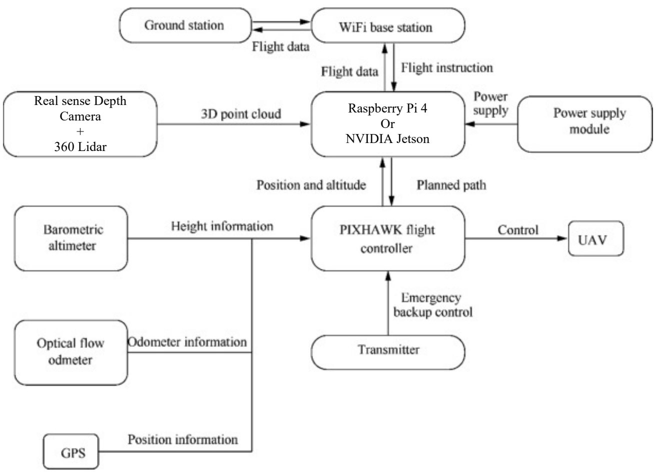


Fig. 2 Drone Sensors Architecture

D. Algorithms

Many methods have been published in the literature to find the best path. Every planning challenge entails a series of decisions that are implemented over time. Furthermore, it is essential to describe in the planning formulation how the state changes as actions are taken. The initial and goal states are included in each path planning process. There are often two types of planning considerations. The first is feasibility: design a strategy that leads to the robot arriving at a desired state, regardless of how efficient it is. The second goal is to design a workable strategy that improves performance in a specific way. However, some optimal criterions were defined to compare the performance of these algorithms: computational time, length of path[6].

a) *A Star Search Algorithm:* There is a well-known and fundamental heuristic method called a star search (A*, A-star, or A* search) [8]. Methodically, it is attempted to minimize the function, formalized as f(n) = g(n) + h(n) taking into account the links between nodes and edges. In mathematical terms, g(n) is the cost of the beginning point or node, while h(n) is the cost of the remaining journey. h(n) hereby constitutes the heuristic base of the algorithm[7].

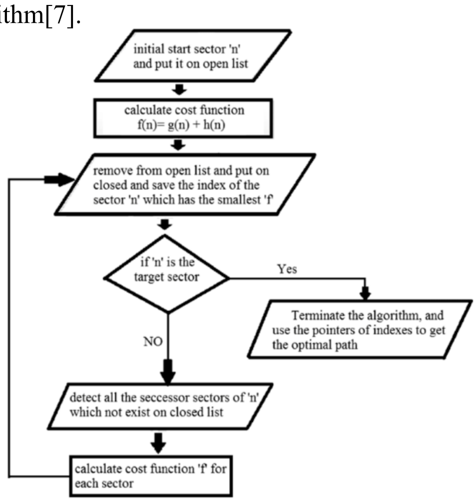


Fig. 3 The A* Algorithm flow chart

b) *RRT Algorithm:* RRT is another probabilistic-based approach that is effective in solving non-holonomic path planning problems in non-convex and high-dimensional domains. It utilizes two functions, tree formation and tree expansion, to construct the tree, which represents the solution space incrementally[9].

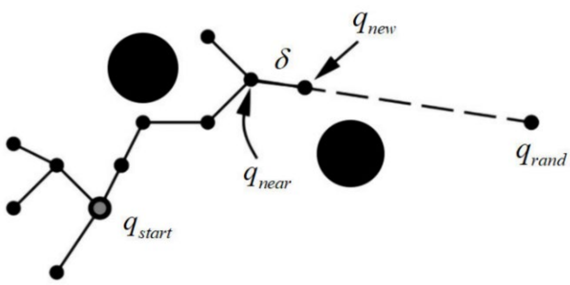


Fig. 4 The RRT Algorithm

c) Ant Colony Optimization Algorithm: Another probabilistic strategy for tackling computing problems that may be simplified to finding optimal pathways via graphs is the Ant Colony optimization algorithm (ACO). Multi-agent approaches inspired by the behaviour of actual ants are referred to as artificial ants. Biological ants employ pheromone-based communication as their primary mode of communication. Combinations of artificial ants and local search algorithms have emerged as the preferred solution for a variety of optimization tasks requiring graphs, such as automobile and internet routing[12].

d) Plant Growth Based Algorithm: The algorithm begins with the seed germ (first bud), while the light source is its target point. The algorithm's essential rules include Phototropism, negative geotropism, apical dominance, and branch development[10]. On a computer, the plant development process is discretized. The plant development behaviour is expected to remain the same for each iteration. Initially, light intensity is calculated, random branches are then calculated, growth vectors are computed, plant growth is computed, and path output is obtained.

```
Algorithm   Plant growth algorithm
Input       3D voxel map
Output      path
    Initialize()
    While !(Reach Target(path)) do
        light intensity ← Light Intensity Calculate(3D voxel map)
        branch ← Random Branch(light intensity)
        growth vector ← Growth Vector Calculate()
        Plant Grow(growth vector)
    path ← Path Clip(path)
```

Fig 5 The Plant growth algorithm

E. Simulation Parameters

ALGORITHM PARAMETERS	
α - Safety Distance	0.8m
β - Safety Distance	0.7m
Velocity safety Distance	1.2m
Critical Safety Distance	0.07m
Max velocity	0.4m/s
Octomap resolution	0.1m
Spherical matrix resolution	6 degrees

0

UAV PARAMETERS	
Mass	1.5kg
Radius	35 cm
Inertia, ixx	0.0348
Inertia, iyy	0.0459
Inertia, izz	0.0977

III. SIMULATION AND RESULTS

After getting a broad grasp of how these algorithms function theoretically, it's crucial to put them to the test on a real drone to determine their effectiveness and usefulness. ROS includes a real-time physics simulator environment called Gazebo[], allowing a drone model to be thoroughly evaluated before a prototype is built. In addition, the drone model is integrated into the Gazebo using a technique known as 'Software in the Loop,' or SITL, which feeds real-life drone data to the physical environment for simulation.

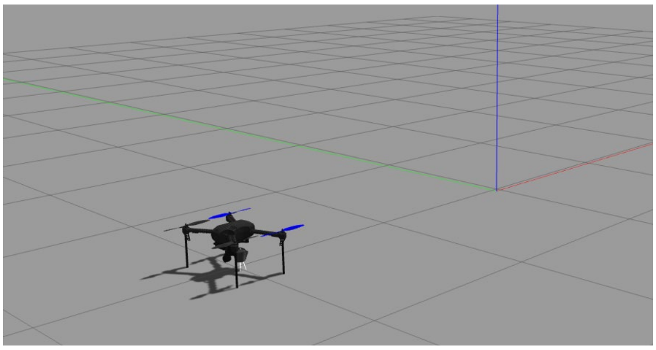


Fig 6 Drone spawned in Gazebo using SITL

To put the various path planning algorithms to the test, you must first examine how they plan the path when used in a variety of challenging situations. The drone created by Gazebo is given particular waypoints with obstacles in between to see how different planners will organise their voyage. As the primary path planner, the most effective path planner will be chosen.

The following test cases were analysed with different obstacles in each scenario. The initial point and target point for the different algorithms were the same, and the computation time and path length planned were noted.

Special Case 1: Wall in front

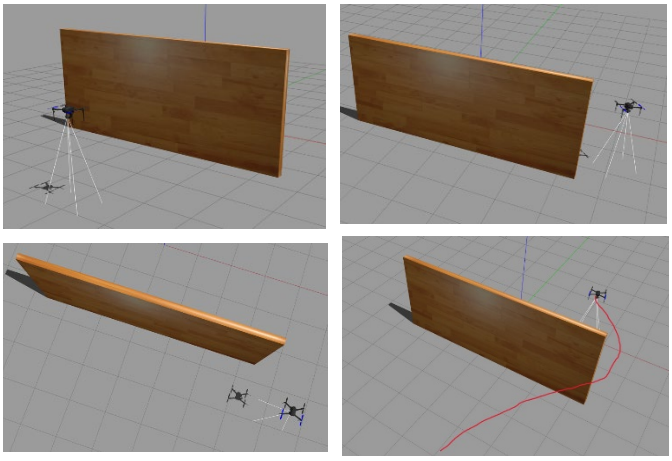
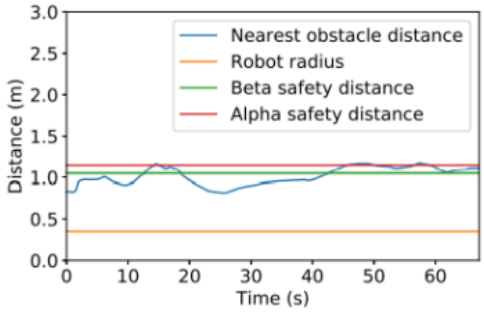


Fig 7 Test scenarios created in Gazebo simulator

Algorithm	**Number of Readings**	**Planning time (s)**	**Average Planning path Length (m)**
Plant Growth	10	1.49	12.7380
A*	10	17.72	11.5950



Special Case 2: Between two walls

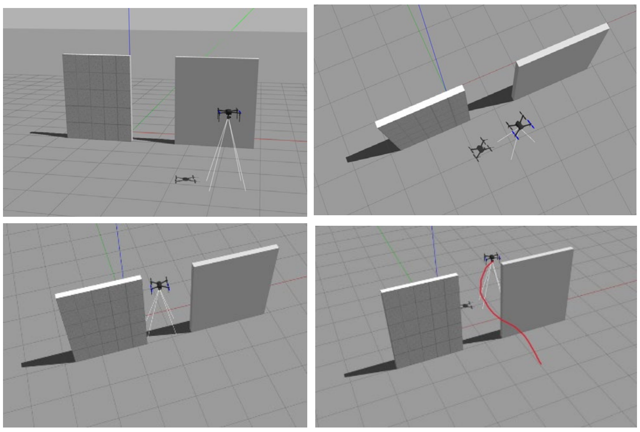
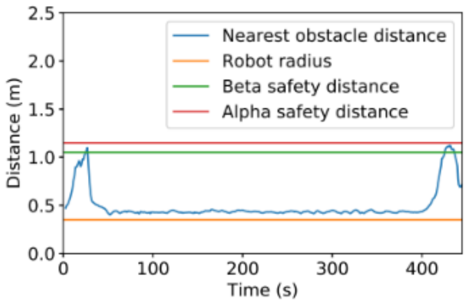


Fig 8 Test scenario 2 created in Gazebo simulator

Special Case 2: Multiple non-stationary obstacles

Algorithm	Number of Readings	Planning time (s)	Average Planning path Length (m)
Plant Growth	10	1.84	13.5864
A*	10	8.73	11.5873



The experiments reveal that the A* algorithm takes the shortest path, however it uses significantly more computing power than the other algorithms. Despite the fact that the Plant Growth algorithm is a relatively new method of path planning, it has showed tremendous promise.

IV. Conclusion

The proposed research was in the view to develop a drone system that is capable enough to smartly navigate into the given waypoints, and efficiently follow the path provided by an optimized planning algorithm. This study included a comparison of multiple algorithms performed with the drone in the Gazebo Simulator using the ROS framework, and it was discovered that the plant growth algorithm produced better results compared to A-star algorithm.

V. References

[1] Johvany Gustave, Jamy Chahal and Assia Belbachir, “Functional Architecture using ROS for Autonomous UAVs,”2020
[2] Pablo Marin-Plaza,Ahmed Hussein,David Martin, and Arturo de la Escalera, “Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles,” 2018
[3] Claudio Sciortino, Adriano Fagiolini, “ROS/Gazebo-based Simulation of Quadcopter Aircrafts,” September 2018
[4] Yadira Quiñonez, Fernando Barrera, Ian Bugueño, Juan Bekios-Calfa, “Simulation and path planning for quadcopter obstacle avoidance in indoor environments using the ROS framework,” January 2018
[5] Emad Ebeid Martin Skriver Kristian Husum Terkildsen Kjeld Jensen Ulrik Pagh Schultz, “A survey of Open-Source UAV flight controllers and flight simulators,”2018
[6] Mehmet Korkmaz, Akif Durdu, “ Comparison of optical path planning algorithms,” TCSET-2018
[7] Z. Batik Garipa, D. Karayelb, S.S. Ozkanb and G. Atalic, “Path Planning for Multiple Mobile Robots Using A* Algorithm,”2017
[8] Issa Mtanos Zidane and Khalil Ibrahim “Wavefront and A-Star algorithms for mobile robot path planning”2017
[9]Chengren Yuan, Wenqun Zhang, Guifeng Liu, Xinglong Pan, Xiaohu Liu ,“A Heuristic Rapidly-Exploring Random Trees Method for Manipulator Motion Planning”2019
[10]Yaoming Zhou, Yongchao Wang, Xuzhi Chen, Lei Zhang & Kan Wu, “A novel path planning algorithm based on plant growth mechanism”2017
[11] AND FANG LI, JIA TAN, PING ZHANG, LANXIANG ZHENG, "The Obstacle Detection Method of UAV Based on 2D Lidar", 2019
[12] Chao Zhang, Ziyang Zhen, Daobo Wang, Meng Li, “UAV path planning method based on ant colony optimization”,2010