BTP PHASE II

IIT PALAKKAD

Obstacle Avoidance Of Quadrotor

Presented By
Shefali Shekhar
(122001043)

Under the guidance of
Dr.Sneha gajbhiye

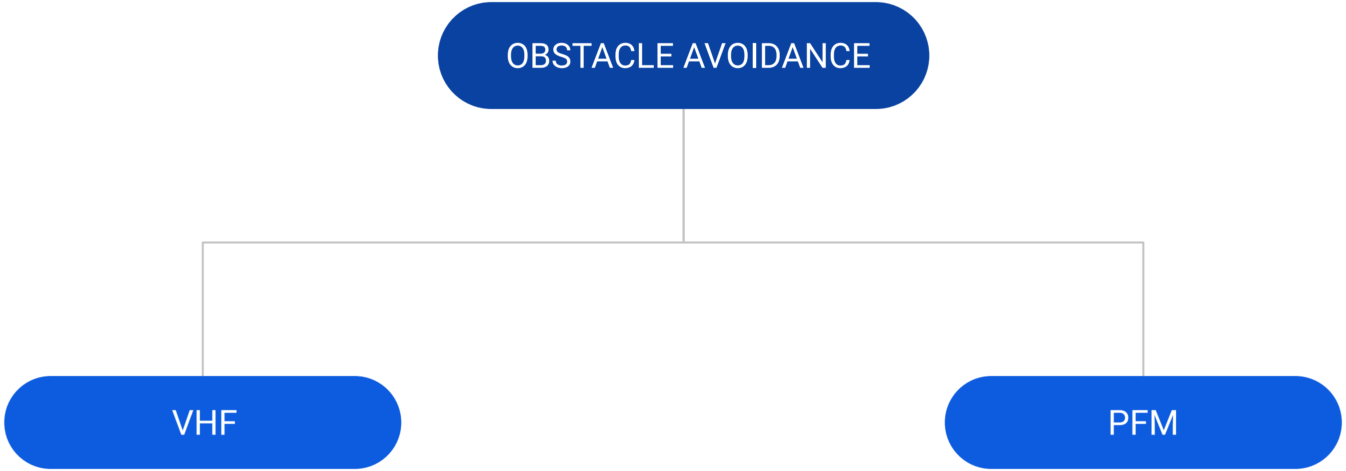
Outline

- Problem Statement
- Solution
- Introduction
- Prior work to Now
- Literature Review
- Quadrotor Configuration
- Lidar Introduction
- Path Planning
- Obstacle avoidance
- Code
- Results
- Conclusion
- Future Work
- References

Problem Statement

Design an **algorithm** for a quadrotor that enables it to follow a set trajectory while dynamically avoiding obstacles. It must detect obstacles in real-time and adjust the flight path to maintain safety and stability. The system should ensure smooth trajectory tracking and operate under various environmental conditions.

Solution



Introduction and Application

What is Quadrotor?

A quadrotor, also known as a quadcopter, is a type of drone or unmanned aerial vehicle (UAV) with four rotors.

These rotors allow for precise control of flight dynamics, enabling stable hovering, vertical takeoff and landing, and multidirectional movement.

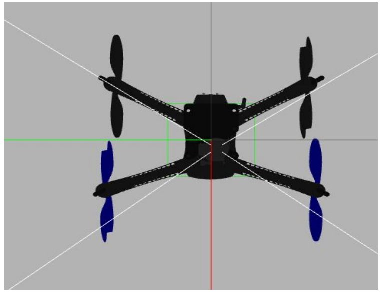


Figure 1:Quadrotor Model

Applications of Quadrotor:

SEARCH AND RESCUE

Figure 2:Search And Rescue

AGRICULTURE

Figure 3:Agriculture

SECURITY AND SURVEILLANCE

Figure 4:SECURITY AND SURVEILLANCE

Src of: fig (2.1)https://s1.cdn.autoevolution.com/images/news/this-life-saving-technology-takes-search-and-rescue-drones-to-the-next-level-181775_1.jpg
Src of: fig (2.2)https://t3.ftcdn.net/jpg/07/66/12/16/360_F_766121645_Erw5tTnBPRvKah1tJO6YlDg6iO1TdfBM.jpg
Src of: fig (3)https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRB-8646v0P03wIg7bCAjwZErR5FZfS_F_qz0xy-b8XQw&s
Src of: fig (4)https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS4yhK-Yx29GO3dd75OskRcNlHWy9u5CpWNc6RoO-0WuA&s

Prior work

Developed a model and conducted simulations for the obstacle

avoidance-driven autonomous navigation of a quadrotor system using MATLAB Simulink.

For a 2D.

Literature Review

Obstacle avoidance is crucial in robotics, ensuring robots navigate without collisions. A key reference is:

- Ribeiro, M. I. (2005). Obstacle Avoidance. Technical University of Lisbon.

Our quadrotor model Configuration:

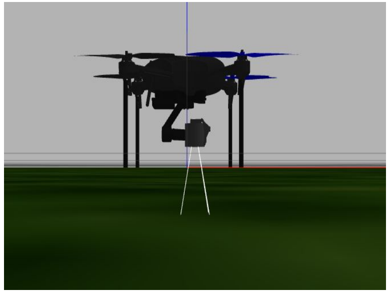


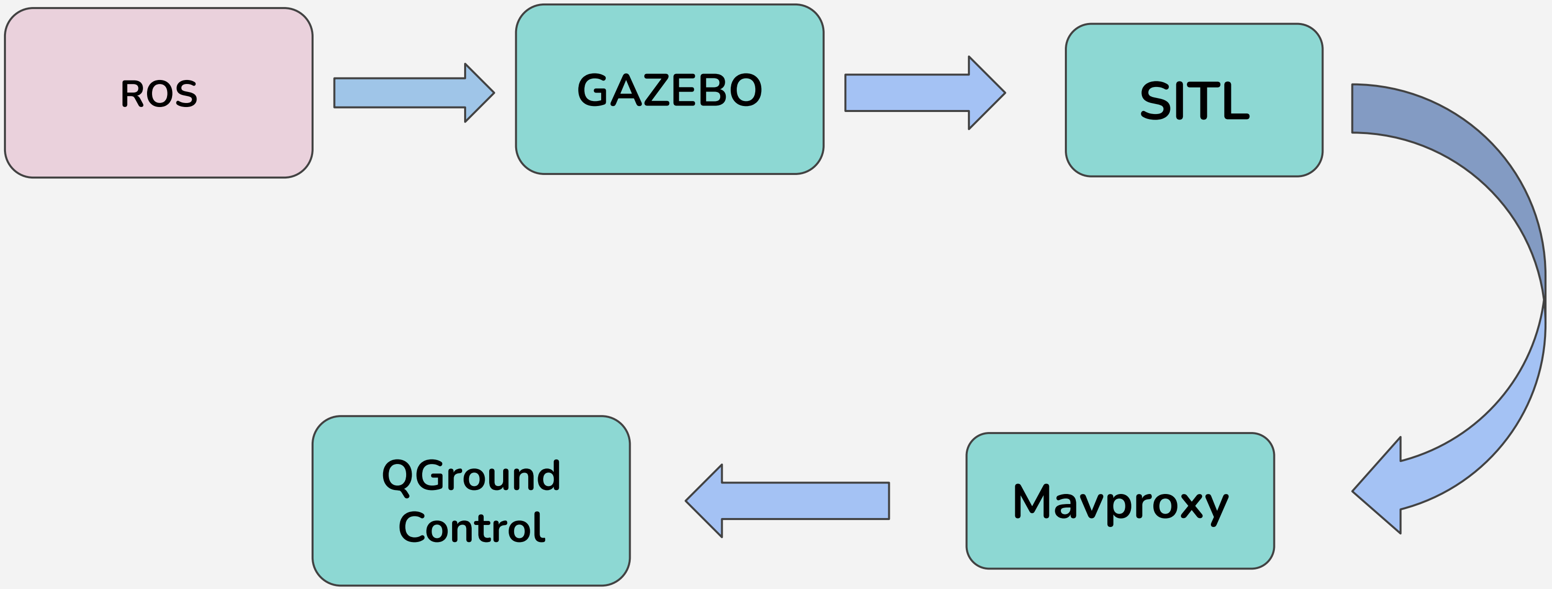
Figure 5:Quadrotor Model

1. Position:
   X: 0m
   Y: 0m
   Z: 0.182466m

Mass: 1.316 kg
Inertia Matrix:
Ixx: 0.0128 kg·m²
Ixy: 0 kg·m²
Ixz: 0 kg·m²
Iyy: 0.0128 kg·m²
Iyz: 0 kg·m²
Izz: 0.0218 kg·m²

- These values represent the physical properties and position of the quadrotor model in the simulation.

Implementation Flowchart



Path Planning

We've employed the waypoint method to plan the drone's path, enabling it to execute shapes such as ***square and circle***. We've validated this approach using **QGroundControl**.

Figure 6.1 : Initial stage of circular Trajectory

Figure 6.2 :Halfway of circular Trajectory

Figure 6.3 : Quadrotor following circular path

Figure 7.1 : Initial stage of square Trajectory

Figure 7.2 : Halfway of Square Trajectory

Figure 7.3 : Quadrotor following square path

LIDAR(Light Detection and Ranging**)**

What is LIDAR?

LiDAR is a **sensor** ,on a drone or quadrotor that uses laser beams to scan the surroundings, enabling the detection of obstacles and creating 3D representations of the environment. This data aids in navigation, obstacle avoidance, and mapping within a Gazebo simulation environment.

Adding 2D LIDAR to our Drone Model

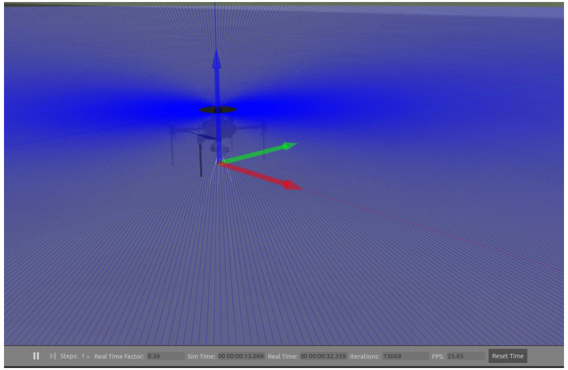
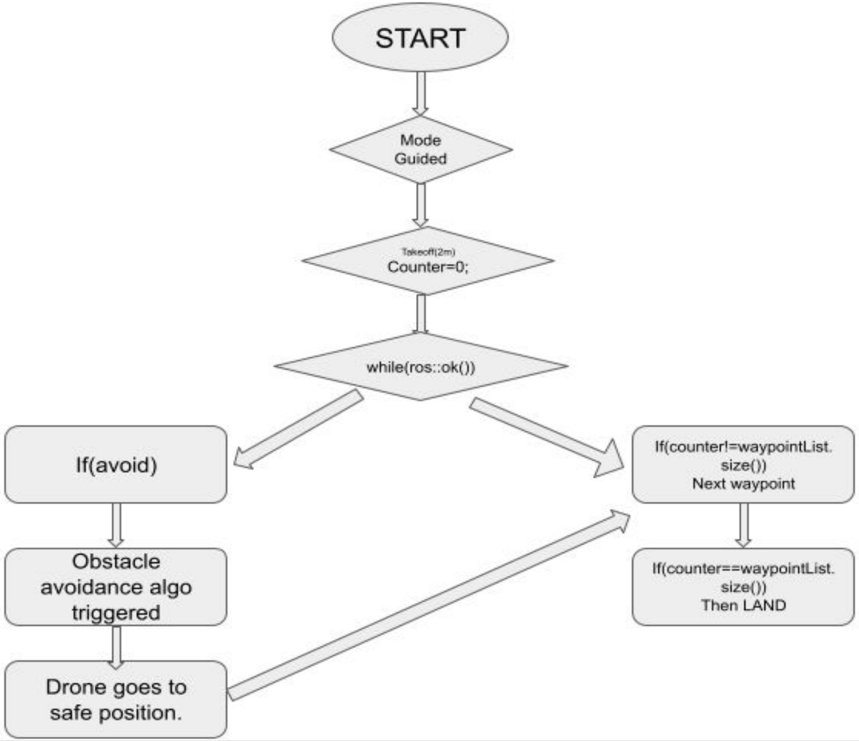


Figure 8 : Quadrotor with LIDAR

The LiDAR is set up to perform a full scan with high sampling and updates 10 times per second. It has a considerable detection range and is integrated with **ROS** through the given plugin and topic.

Obstacle Avoidance Algorithm



PFM

Path planning using artificial potential fields is based on a simple and powerful principle.

Robot as positive charge(+)
Obstacle as positive charge(+)
Goal as negative charge(-)

$$U_{repi}(q) = \begin{cases} \frac{1}{2}\, k_{obsti}\, (1/\, d_{obsti} - 1/\, d_0)^2 & \text{If } d_{obsti}(q) < d_0 \\ 0 & \text{If } d_{obsti}(q) \geq d_0 \end{cases}$$

where $d_{obsti}(q)$ is the minimal distance from q to the obstacle i, k_{obsti} is a scaling constant and d_0 is the obstacle influence threshold.

Including Obstacle In our Environment

- Using bus as obstacle in our environment

X	11.783400
Y	-0.295179
Z	0.00
Roll	0.00
Pitch	0.00
Yaw	0.00

Table 1 : Bus configuration after obstacle

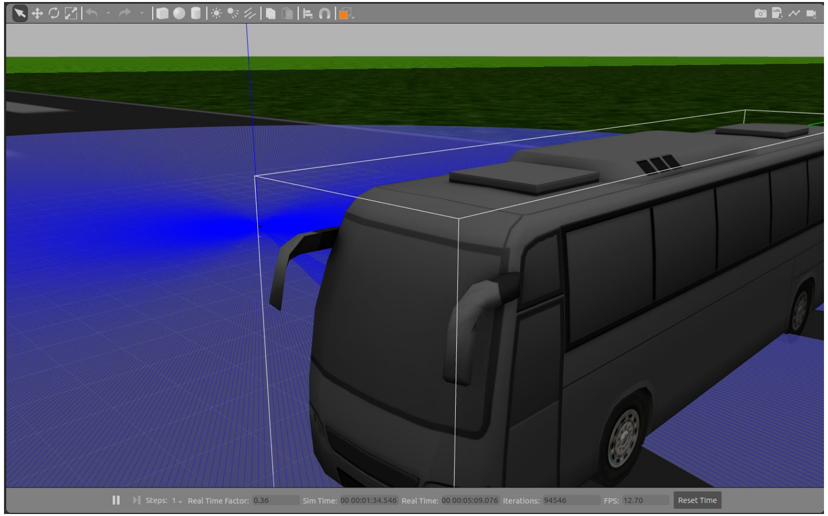


Figure 12 : Obstacle

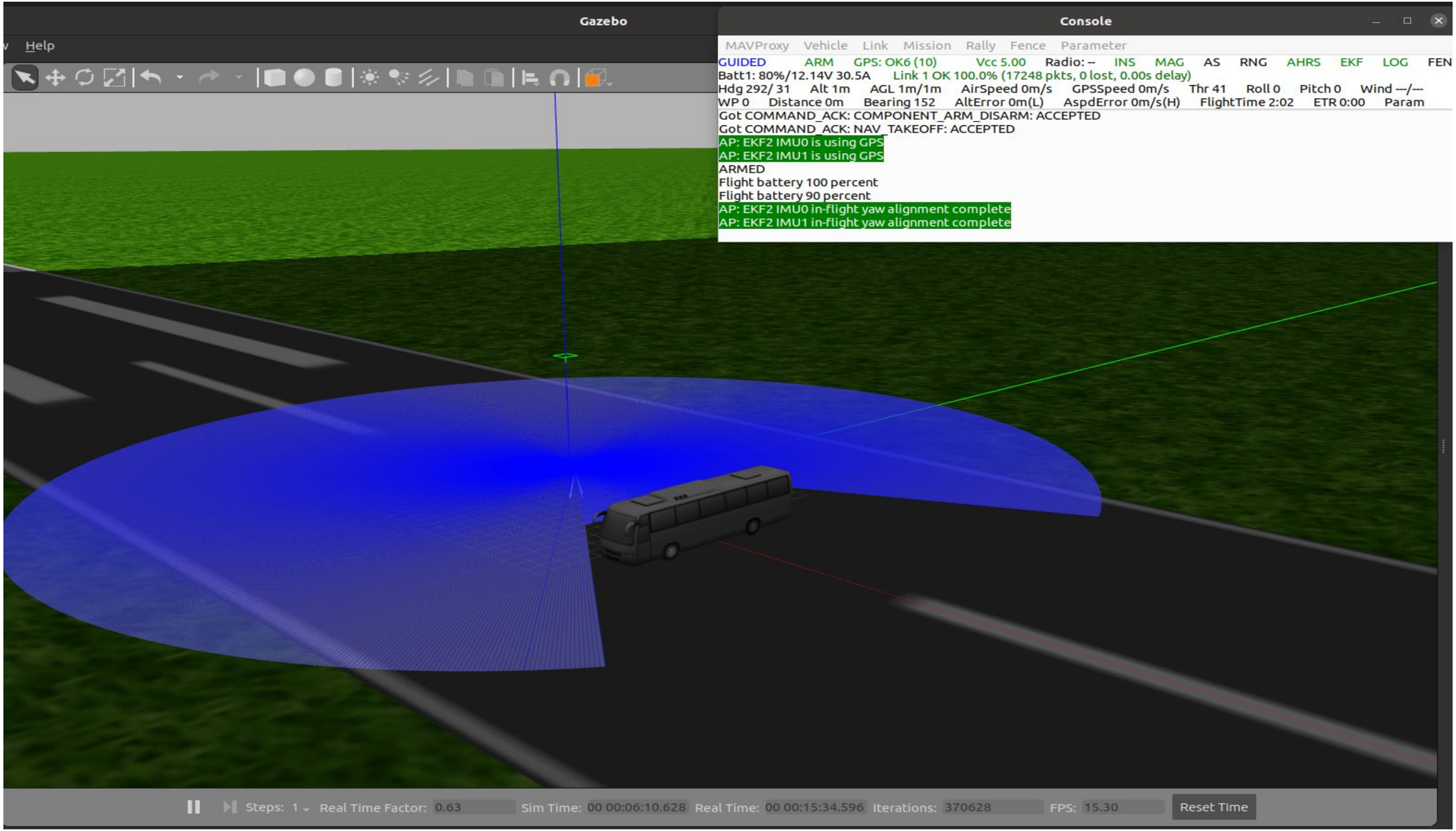


Figure 9 - Drone takeoff from origin(0,0,0)

```
while(ros::ok())
    {
        ros::spinOnce();
        rate.sleep();


        if(check_waypoint_reached(.3) == 1)
        {
            if(t>0){
            counter--;
            t=0;
            }
            if (counter < waypointList.size())
            {
 set_destination(waypointList[counter].x,waypointList[counter].y,waypointList[counter].z,
        waypointList[counter].psi);
                counter++;
            }else{
                //land after all waypoints are reached
                land();
            }
        }

            }

    return 0;
  }
```

Quadrotor following the square path

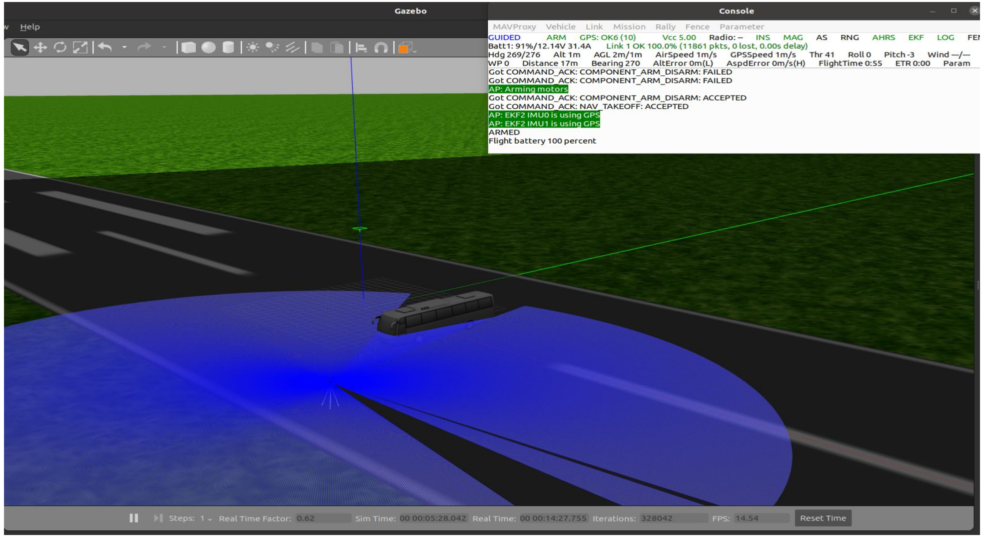


Figure 10 : Quadrotor following square trajectory

```
sensor_msgs::LaserScan current_2D_scan;
current_2D_scan = *msg;
float avoidance_vector_x = 0;
float avoidance_vector_y = 0;
bool avoid = false;

for(int i=1; i<current_2D_scan.ranges.size(); i++)
{
        float d0 = 3;
        float k = 0.5;

        if(current_2D_scan.ranges[i] < d0 && current_2D_scan.ranges[i] > .35)
        {
                avoid = true;
                float x = cos(current_2D_scan.angle_increment*i);
                float y = sin(current_2D_scan.angle_increment*i);
                float U = -.5*k*pow(((1/current_2D_scan.ranges[i]) - (1/d0)), 2);

                avoidance_vector_x = avoidance_vector_x + x*U;
                avoidance_vector_y = avoidance_vector_y + y*U;

        }
}
```

Avoiding the obstacle using obstacle avoidance algorithm

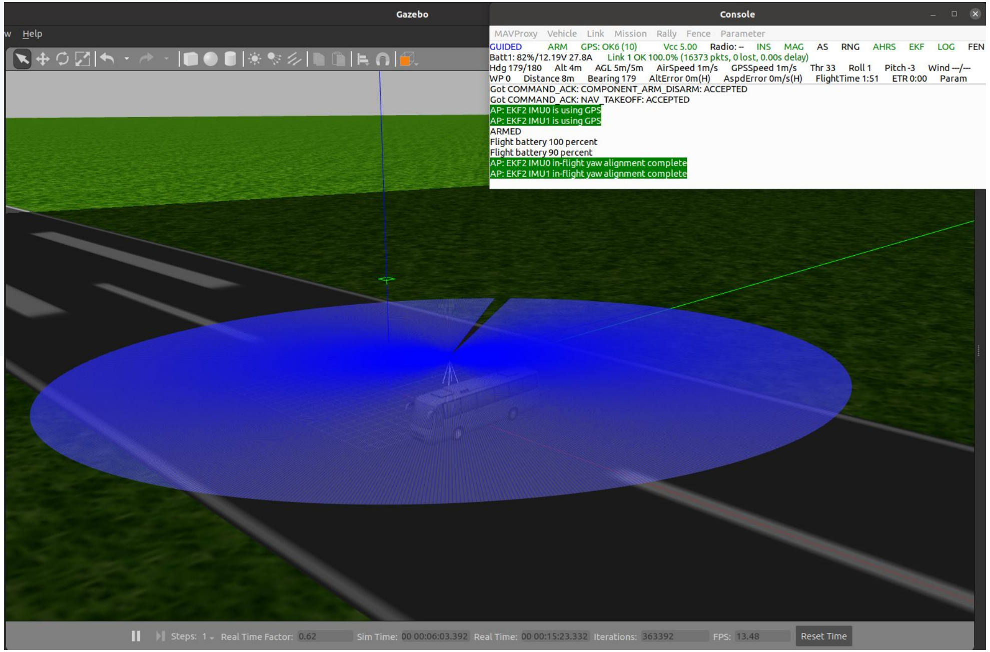


Figure 11 : Quadrotor avoiding Bus

Results

We're conducting a comparison between the quadrotor's coordinates in two simulations. One simulation includes obstacle, while the other doesn't. By analyzing these scenarios, we aim to understand how obstacles affect the quadrotor's movement and position.

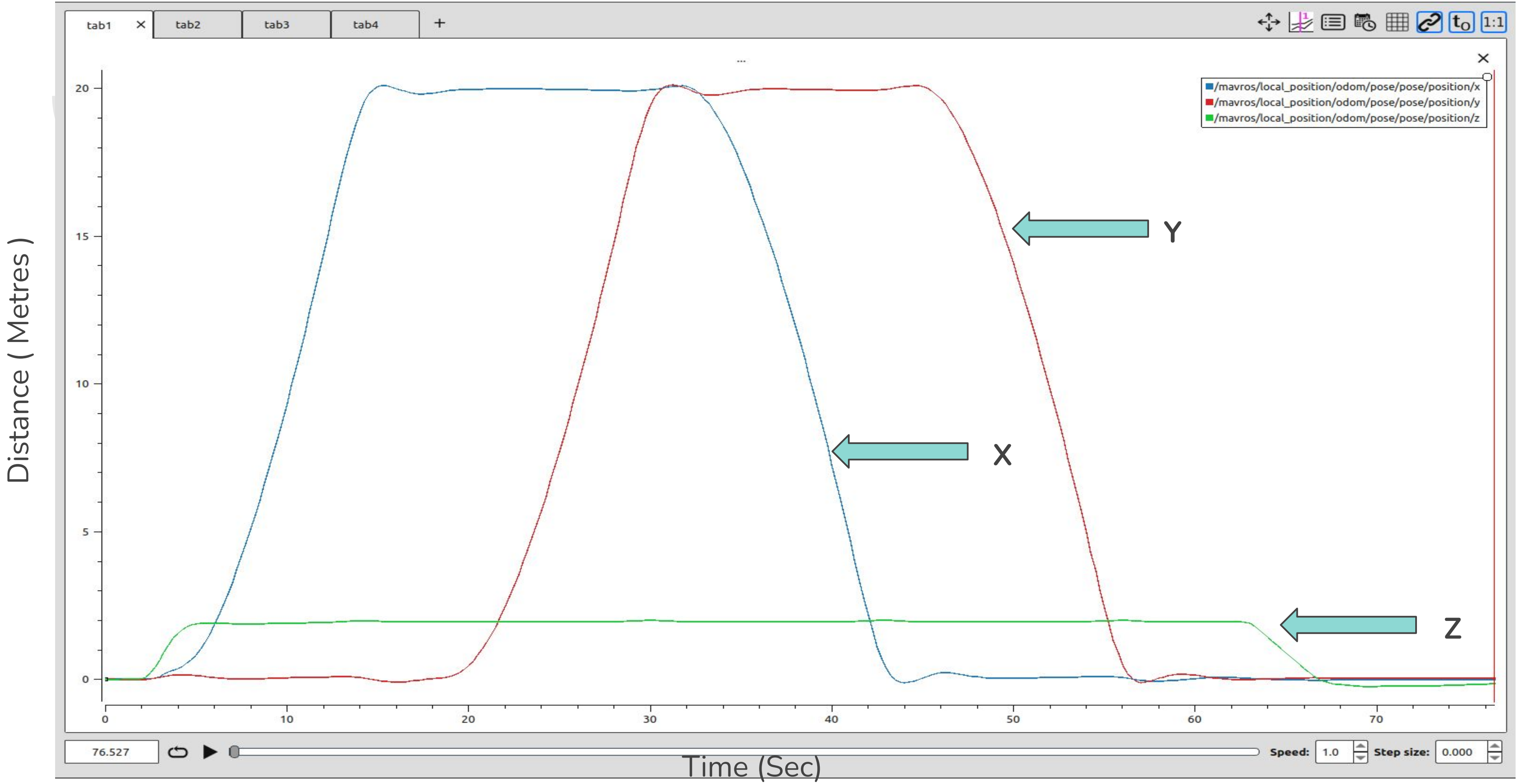


Figure 13 - X,Y,Z position of Quadrotor following square path **without** obstacle

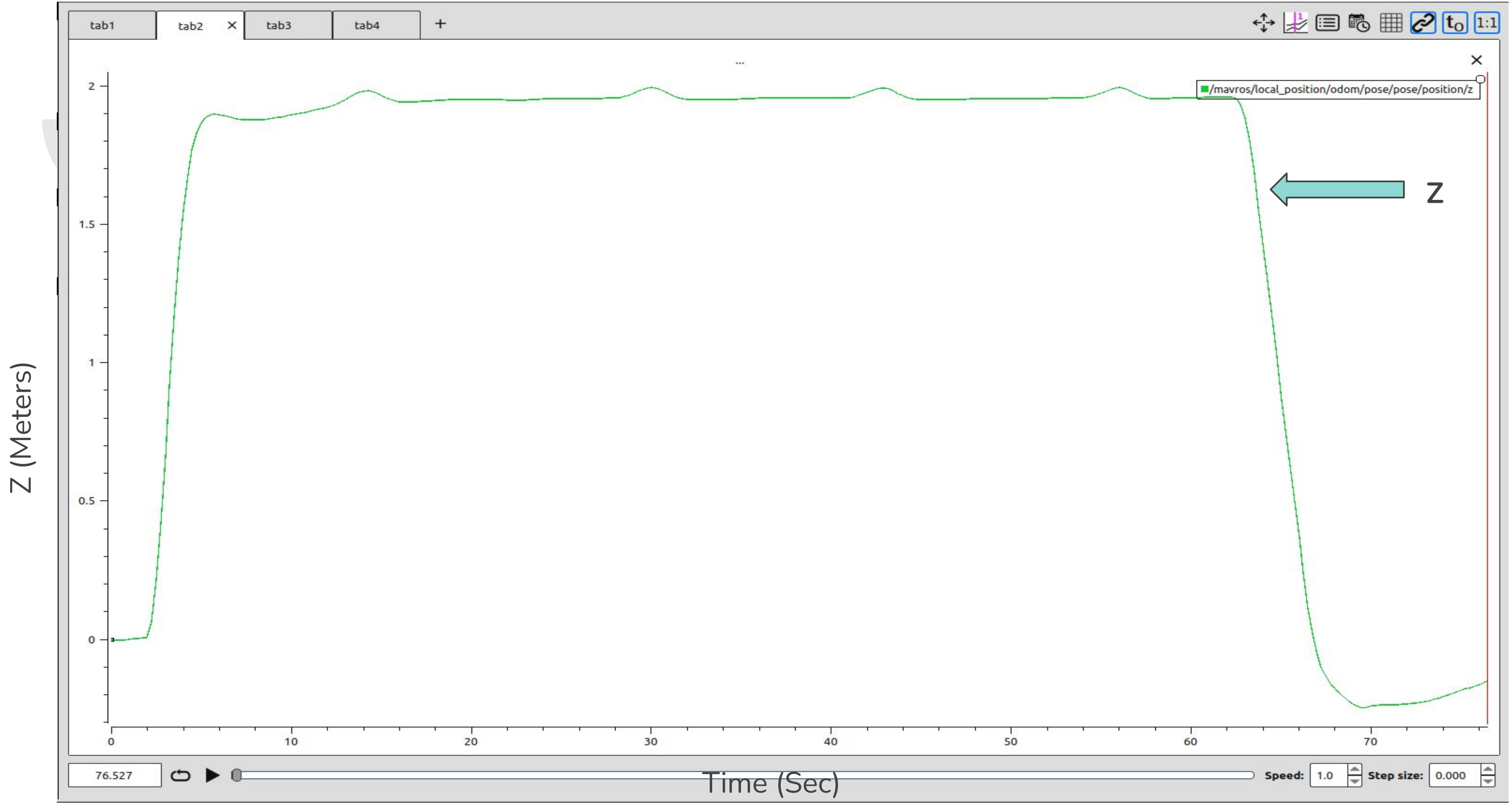


Figure 14 - Z position of Quadrotor following square path **without** obstacle

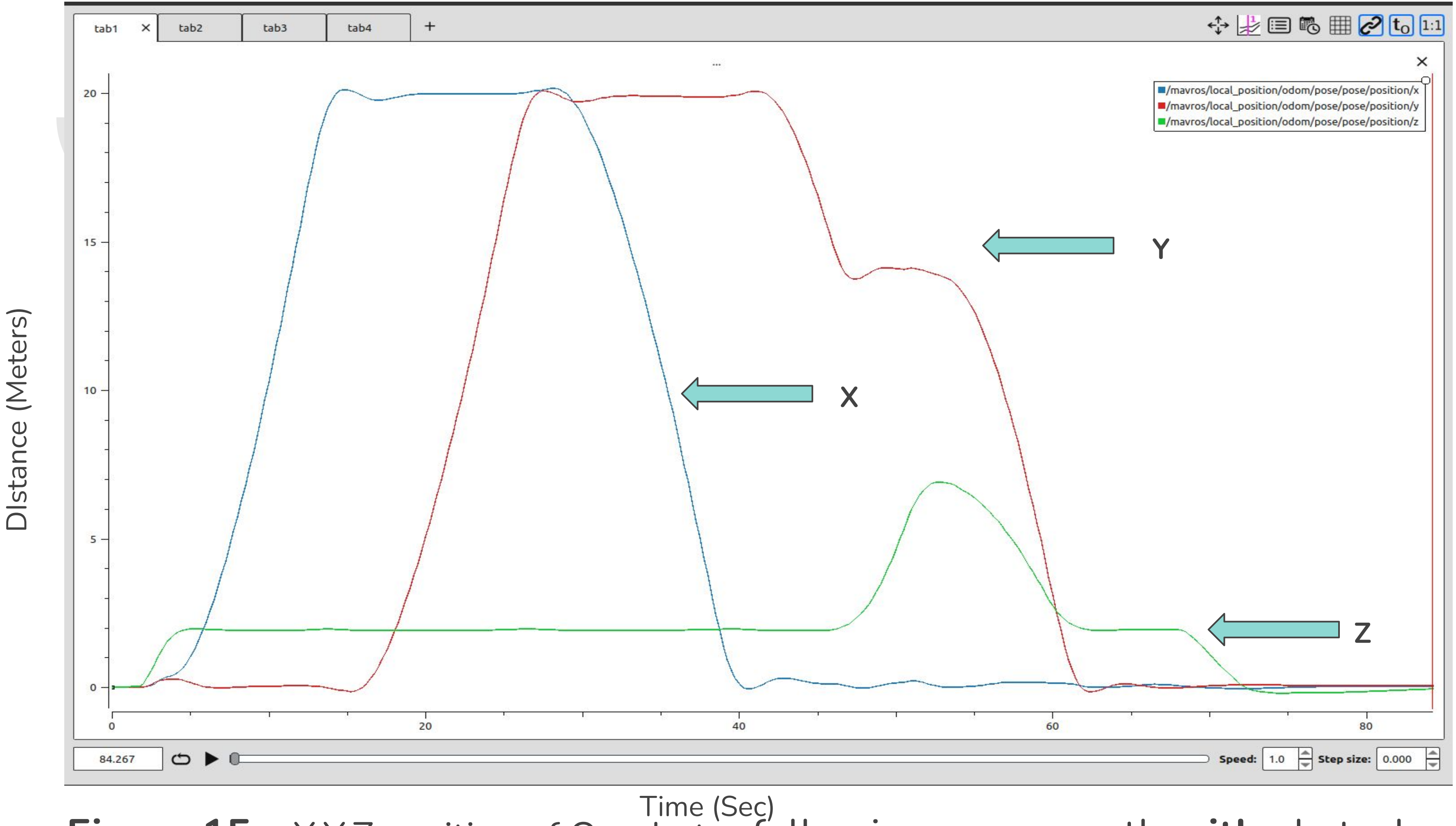


Figure 15 - X,Y,Z position of Quadrotor following square path **with** obstacle

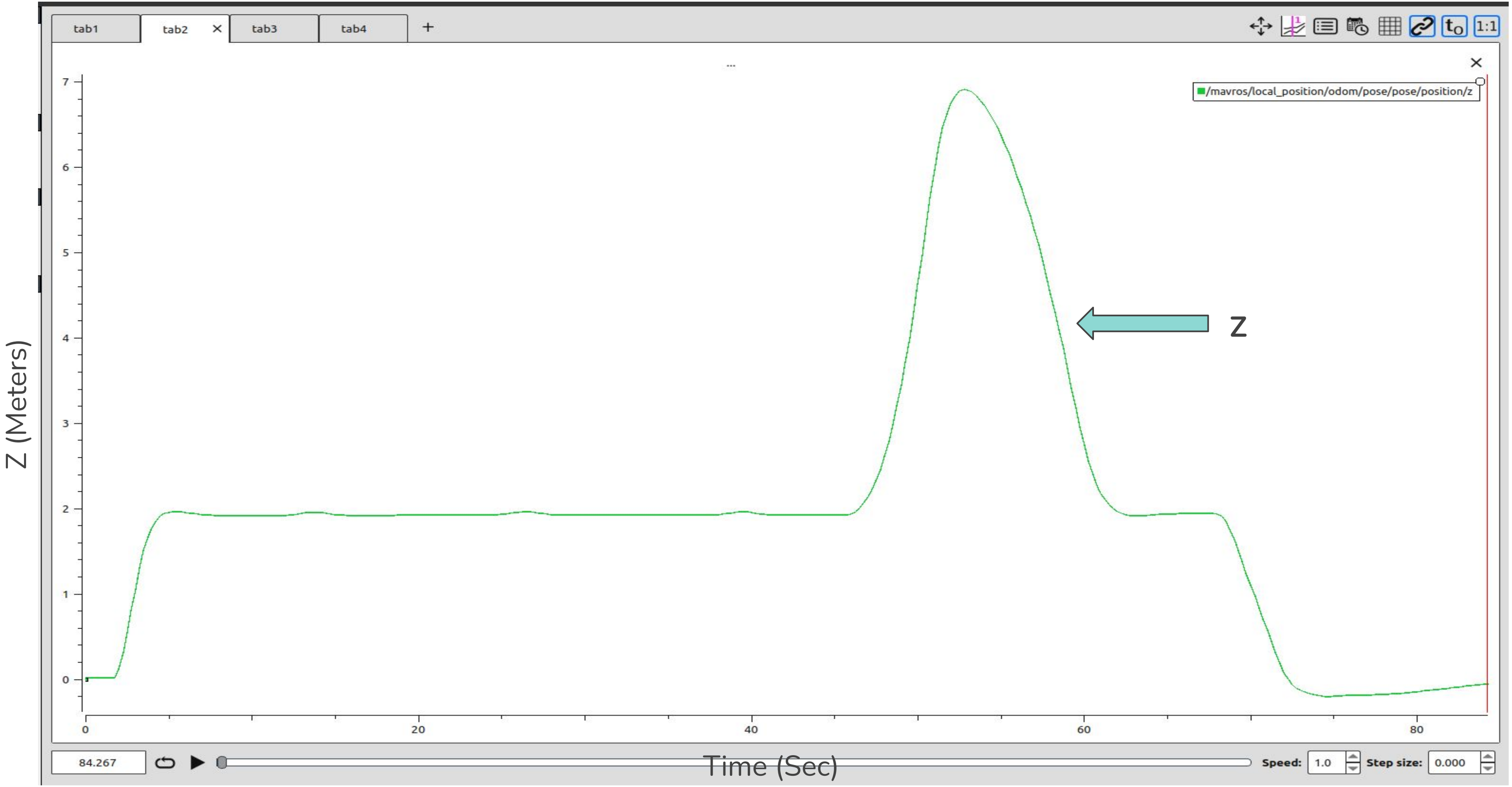


Figure 16 - Z position of Quadrotor following square path **with** obstacle

Conclusion

- Developed a way for drones to avoid obstacles using a special laser sensor and a smart algorithm.
- Put it to the test and got some results, but there's room to make it even better.
- Spent a lot of time figuring out how to make the drone see obstacles using computer simulations.
- Showed that our new obstacle-detecting method is simple and works well for drones, using data we collected to prove it.

Future work

- Expand research by introducing more obstacles to analyze UAV behavior in varied environments.
- Implement a secondary strategy where the UAV ascends upon detecting closely spaced obstacles, reducing collision risks.
- Introduce climbing to a safe altitude when the UAV encounters closely positioned obstacles, enhancing adaptability in complex environments.
- Enhance algorithm complexity by integrating additional obstacles, enabling more efficient navigation in intricate environments and prompting further exploration of advanced multi-obstacle avoidance methods.

References

- O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” International Journal of Robotics Research, vol. 5, no. 1, pp. 90–98, 1995.

- Molina, José. (2022). "Simulation in real conditions of navigation and obstacle avoidance with PX4/Gazebo platform. Personal and Ubiquitous Computing." 26. 10.1007/s00779-019-01356-4.

- M. Arshad, J. Ahmed, and H. Bang, "Quadrotor Path Planning and Polynomial Trajectory Generation Using Quadratic Programming for Indoor Environments," vol. 7, pp. 122, February 2023.

Video


THANK
YOU !