DRONE SIMULATION

Project Report

EKLAVYA MENTORSHIP PROGRAMME

At

SOCIETY OF ROBOTICS AND AUTOMATION,
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
MUMBAI

JUNE 2020

ACKNOWLEDGMENT

MENTORS:
NEHA
OMKAR S
LUKESH

MEMBER 1:
PRACHITI BARGE

MEMBER 2:
PRERNA JAGESIA

MEMBER 3:
SAMINA ATTARI

MEMBER 4:
KHUSHI BARJATIA

TABLE OF CONTENTS

1. PROJECT OVERVIEW
2. INTRODUCTION
3. APPROACH TOWARDS OUR PROJECT
4. WORKFLOW OF OUR PROJECT
5. PROCEDURE
6. TRIALS AND ERRORS
7. EFFICIENCY AND ACCURACY OBTAINED
8. FUTURE ASPECTS OF OUR PROJECT
9. REFERENCES

PROJECT OVERVIEW:-

A Drone or Quadcopter is a Vehicle that has a large potential for performing tasks that are dangerous and costly for humans. Examples are inspection of high structures, humanitarian purposes or search and rescue missions . Quadcopters generally have two rotors spinning clockwise (CW) and other two rotors spinning counterclockwise (CCW). Flight control is provided by independent variation of the speed and hence lift and torque of each rotor. Pitch and roll are controlled by varying the net centre of thrust by varying the net torque. Quadrotors do not usually have cyclic pitch control, in which the angle of the blades varies dynamically as they turn around the rotor hub.

REASON OF CHOOSING OUR PROJECT AS OBSTACLE AVOIDANCE IN A SIMULATED QUADCOPTER-

Quadcopter drones which are becoming increasingly popular are used to capture video for promotional and surveillance purposes. The main reason to have obstacle avoidance is to prevent the drone from crashing, either through pilot error or when engaged in autonomous flight. One major bottleneck to achieving full integration of unmanned aviation is collision avoidance . For this we have simulated a drone and performed obstacle avoidance on it.

SOFTWARE USED IN OUR PROJECT-

ROS :

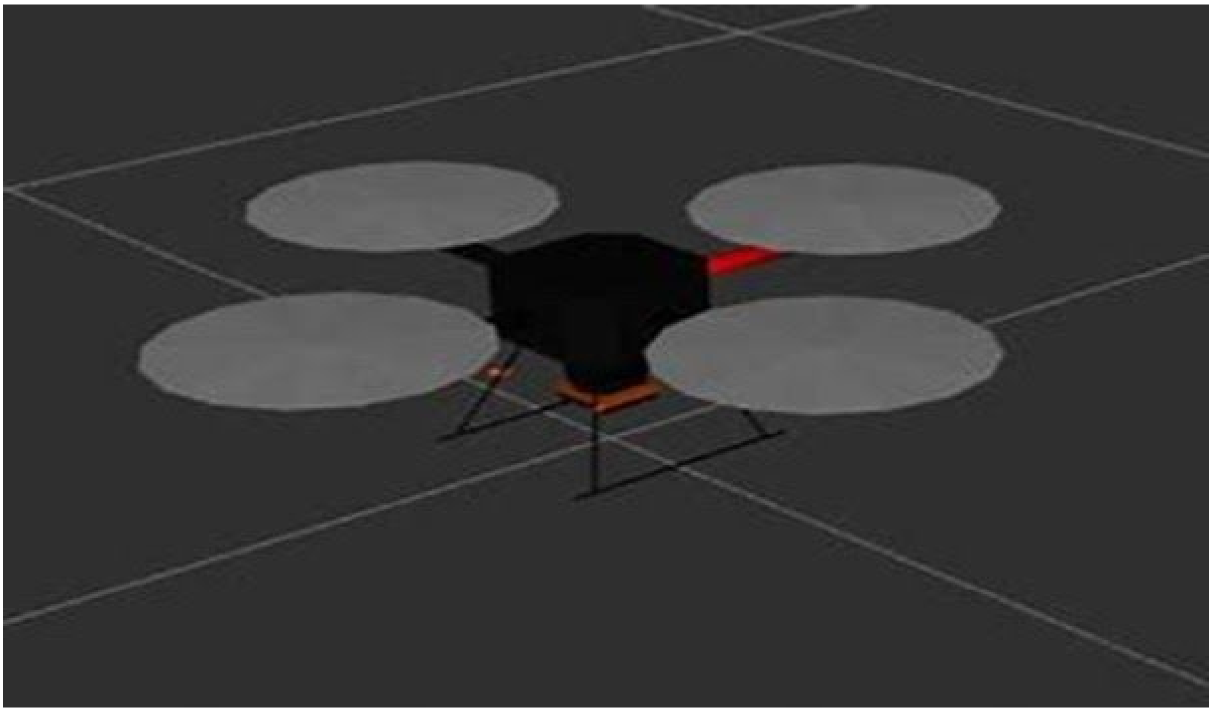
For simulation of the drone we have used software like Ros and Gazebo . ROS is a Robot Operating System and it is a robotics framework which provides tools and libraries to help software developers to create robot applications .ROS has a wide collection of libraries , tools, and conformity with powerful programs like Gazebo . Ros consists of stacks , packages , nodes , topics , messages , services , header, bags, master, manifests and parameter server.

GAZEBO:

For simulation of the drone we have used Gazebo which is a simulation software. Gazebo is a powerful 3D simulation environment for autonomous robots that is particularly suitable for testing object-avoidance and computer vision.

BRIEF EXPLANATION ON THE GOALS WE AIM TO ACHIEVE THROUGH OUR PROJECT-

We have simulated a hector quadrotor which has a lidar sensor HOKUYO 4pn 30lx to perform obstacle avoidance .Lidar sensor gives a proper field of view of range -135 degrees to 135 degrees. It is more convenient and efficient to use a lidar sensor in comparison with a sonar sensor. In our project of drone simulation, the lidar sensor can detect the obstacle and thus avoid it from a distance ranging from 0.3m to 30m in length. The maximum height to which our drone can fly is 3m.We have also used the sonar sensor which helps in take-off and landing of the drone. Our other goal which we have achieved through our project is goal to goal wherein we have used a code to simulate the drone to fly from one particular destination to another. Obstacle avoidance and flying of the drone from one particular destination to another are two most basic and useful tasks which a drone must perform.

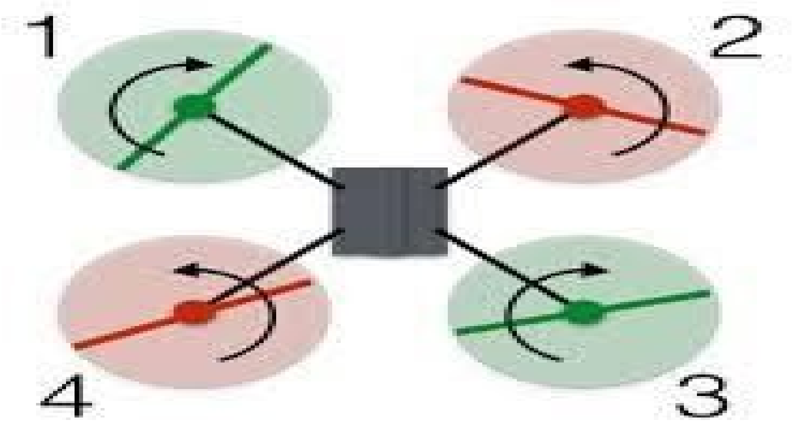


HECTOR QUADROTOR MODEL IN GAZEBO (The one we used)

INTRODUCTION:-

DRONE THEORY-

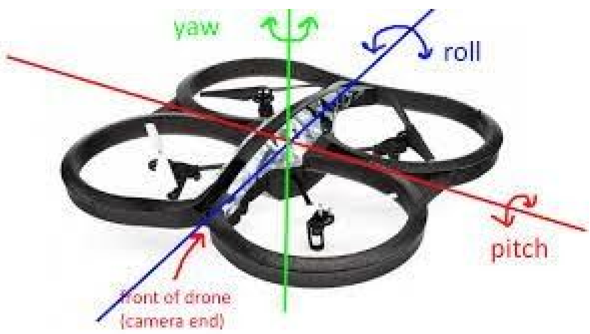
When it comes to flying a drone in practical situations, the first and the most important part of it is controlling the speed of motors in order to give the drone a desired direction. For a quadcopter drone specifically, an opposite pair of motors rotate in clockwise orientation while the other two rotate in counter-clockwise orientation, this is mainly because through this configuration the net torque and the net angular momentum gets cancelled and hence the drone will not tumble around when in air.



HOW DO DRONES FLY?

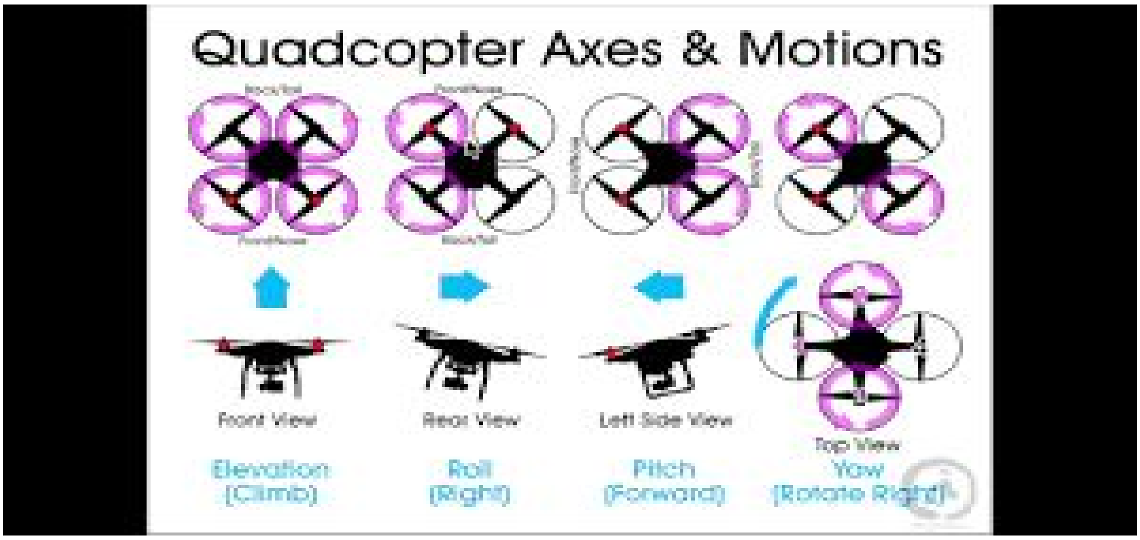
As stated above moment balancing will keep the drone horizontal in air, but there needs to be some force which lifts up the drone otherwise it won't fly, now the spinning blades of the drone push the air down, and as a reaction this air pushes the drone upwards, namely, this force is called thrust that provides lift to the drone and also balances weight of the drone for translational stability. The faster the rotors spin,the greater is the lift and vice-versa. For upward and downward motion of the drone the speed of the motors can be increased and decreased respectively so as to get a net upward or downward force for ascending and descending.

For sideways motion and rotation of the drone about an axis passing through the center of the drone, there are 3 types of movements of a drone- Roll, Pitch, Yaw.



Say for example, if the drone needs to go it's left it roll to one side which can be done by increasing the speed of motors 2,3 and decrease the speed of motors 1,4 so that the drone rolls towards one side, which in turn will generate a horizontal component of thrust now, which will be responsible for drifting the drone to the left, and after it is drifted to it's left a little speed of all the motors can now be increased, to help the drone retain its original configuration and continue further.

All the movements of drone along with motor speed adjustments can be visualized in the figure below-



PID IN DRONES :-

The most important part of controlling any unmanned vehicle is its stability. An unstable flight can lead to crashes and at times cause great damage to the drone, so having a PID controller in a drone is a very important aspect.

What is PID?

P - Acts proportional to the controller sensor signal(Angle error).

I - Keeps on summing up the error angle over time and acts to compensate for this total error.

D -Responds to the rapid change in the angle by taking into account rate of change of error angle.

$$u(t) = kpe(t) + ki(\text{integral of } e(t)dt) + kd(d/dt)e(t)$$

PID is needed in drones for stabilizing it in air as well as to maintain a constant altitude (as in our project). By adjusting the values of kp,kd to optimum and ki to low, optimization in flight can be obtained.

WHY QUADCOPTER?

Quadcopter:-

As already discussed above a quadcopter has 4 motors of which the opposite pair of motors rotate in the same orientation. This particular configuration keeps the drone a stable and hence the chances of the drone crashing or spinning are less.

Tricopter:-

Whereas in a tricopter, since there are odd no. of motors, the torque due to one motor is unbalanced and can cause the tricopter to spin. This problem is encountered by mounting one propeller at a variable angle. This adjustment introduces a yaw servo on the drone which is quite expensive.Also, it has a complex mechanism which requires expensive parts, and if our drone crashes it causes a lot of loss as these parts of the drone are very delicate and easily get damaged, so it requires a lot care to handle them.

Bicopter:-

Firstly, its working is very complex as it will require 2 servo motors, which makes it less stable and difficult to steer. Imagine that you are holding a pencil at the center of its body and trying to balance it sideways, it is very likely that it will roll to one side and fall off, this analogy can be extended to a bicopter also, if the center of gravity is not in line with the center of lift of the propellers, your drone will roll sideways as it is tipped over from the weight.

Thus, considering all things we can say that the quadcopter provides a more accurate and efficient way of handling and maneuvering the drone as compared to others.

SIMULATION:-

A **simulation** is an approximate imitation of the operation of a process or system; that represents its operation over time. It is the process of mathematical modelling, performed on a computer, which is designed to predict the behaviour of or outcome of a real world or a physical system. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. Simulation is also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not yet built, or it may simply not exist.Simulation of a system is represented as the running of the system's model. It can be used to explore and gain new insights into new technology and to estimate the performance of systems too complex for analytical solutions.

Robot simulation is an essential tool in every roboticist's toolbox. A well-designed simulator makes it possible to rapidly test algorithms, design robots, and perform regression testing using realistic scenarios.

There are mainly 26 types of different robotics simulation softwares, some of them are listed here-

1. GAZEBO

Gazebo is a multi-robot simulator with support for a wide range of sensors and objects. The software is ROS compatible along with many other Willow Garage robotics platforms. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. At your fingertips is a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Best of all, Gazebo is free with a vibrant community.

2. V-REP

Designed by Coppelia Robotics, V-rep is one of the most advanced 3D simulators for industrial robots. The tool offers support for a wide range of programming languages including C/C++, Python, Java, Lua, Matlab or Urbi. It has support to develop algorithms to simulate automation scenarios, the platform is used in education as well by engineers for remote monitoring or safety double-checking.

3. WETBOTS

Webots is a 3D simulation platform developed by Cyberbotics and used in service and industrial simulations. The tool offers support for Windows, Linux and Apple platforms, and is one of the most used simulation software in education or research purposes. Any robot can be modeled, programmed and simulated in C, C++, Java, Python, Matlab, or URBI. The software is compatible with external libraries like OpenCV.

4. MICROSOFT ROBOTICS DEVELOPER STUDIO

Microsoft Robotics Developer Studio (MRDS) is a free 3D simulation software designed by Microsoft with support for a wide range of sensors and other robotic platforms. The platform is compatible with all Windows OS versions and it could be used with C# Visual Studio 2010 to create new services for RDS. The MRDS offers support to simulate service robots including Kinect technology. In collaboration with Parallax, Microsoft designs a robotic platform called Eddie used for creativity, innovation and experimentation.

Of these we have used Gazebo simulator on the ROS development Studio(RDS) platform.

ABOUT GAZEBO-

Gazebo is an open-source 3D robotics simulator. Gazebo was a component in the Player Project from 2004 through 2011. Gazebo integrated the ODE physics engine, OpenGL rendering, and support code for sensor simulation and actuator control. In 2011, Gazebo became an independent project supported by Willow Garage. In 2012, Open Source Robotics Foundation (OSRF) became the steward of the Gazebo project. OSRF changed its name to Open Robotics in 2018.

Gazebo can use multiple high-performance physics engines, such as ODE, Bullet, etc (the default is ODE). It provides realistic rendering of environments including high-quality lighting, shadows, and textures. It can model sensors that "see" the simulated environment, such as laser range finders, cameras (including wide-angle), Kinect style sensors, etc.

DIFFERENT APPROACHES OF HOW PROJECT CAN BE DONE:-

Using monocular vision:

Obstacle Avoidance can be performed by using images from a single camera.Using the extrinsic camera calibration, mapping of each pixel from the camera frame to a corresponding real world coordinate can be carried out to give a bird's view perspective of the image. This is very important as it helps us distinguish obstacles from the ground. The principle used here is that all objects which really belong to the ground plane are represented by their real shape while the objects which are not on the ground plane are heavily distorted in this top view. So these obstacles can be extracted using the OpenCv library.

After the successful abstraction of obstacles , the pose of the robot relative to the track can be found out and suitable obstacle avoidance algorithms can be used.

PROS:

1)More accuracy and reliability as compared to lidar and sonar sensors. The corners can be detected accurately as opposed to using lidar.
2) Using images gives us an added advantage of flexibility in detecting obstacles. We can decide which objects are to be treated as obstacles and which can be safely ignored.
It also allows us to program our bot to follow a certain lane.

CONS:

1) Image processing requires high computational power and storage.
2)Also the weight of the drone increases due to the use of heavy weight cameras.

OBSTACLE AVOIDANCE THROUGH NAVIGATION AND 2D MAPPING:-

Autonomous navigation is the ability of a drone to plan its route and execute its plan without human intervention. This implies two different problems: knowing how to move towards a target and having the ability to avoid any obstacles along the way.

2D MAPPING AND NAVIGATION-

Once the quadrotor can reliably and stably navigate the environment based on a series of desired waypoints, the quadrotor system can be used to sense and comprehend its surrounding environment. A map is a representation of the environment where the quadrotor is operating. To operate in the map, the quadrotor needs to know its position in the map coordinate frame. Ultimately, the goal is to develop a system that allows the quadrotor to autonomously reach a desired goal state in the map without colliding with obstacles. The path from the initial state to the goal state can be a series of waypoints or actions, if a path exists.

A perfect estimate of the quadrotor's pose needed to map creation is typically not available, especially in indoor environments. Simultaneous Localization and Mapping is the process of both making and updating a map of the environment while also estimating the system's location in that map. This map can be saved and used later for localization and navigation without the need to rebuild the map. Typically, lasers are used to create two dimensional maps based on range measurements. However, these are often too large in size and mass or power intensive for use on small quadrotors. Instead, a RGBD sensor is used as the primary means of measuring the environment. This sensor, a Microsoft Kinect, contains stereo cameras as well as a depth sensor. The Kinect outputs point clouds which can be used by the SLAM algorithms to build the map of the environment. It is also possible to use these point clouds to emulate a 2D laser scan output.

In an ideal world, the map accurately reflects the environment, the environment is stable, and the localized estimate of the quadrotor's pose in that map is accurate. However, in the real world, the environment is dynamic and sensor measurements are noisy. Therefore, obstacle avoidance and detection systems are developed as methods of continually updating the map. The ROS Navigation Stack combines all of these requirements into a complete sense-plan-act system.

APPROACH OF OBSTACLE AVOIDANCE USING 2D MAPPING-

To achieve our goal of obstacle avoidance we need to first create a map . It is important to know where the robot is, to create a map of the given environment, to interact every time with the map and to optimize the route to get a smooth path. The most needed features are the map, the pose of the robot, sensors and a navigation algorithm.

The first essential feature for navigation is the map. Using RViz, the navigation system is equipped with a very accurate map from the time of purchase, and the modified map can be downloaded periodically so that the drone can be driven to the destination based on the map. Like a navigation system, a drone needs a map, so we need to create a map and provide it to the drone, otherwise the drone should be able to create a map by itself. SLAM (Simultaneous Localization and Mapping) is developed to let the drone create a map with or without the help of a human being. This is a method of creating a map while the drone explores the unknown space and detects its surroundings, estimating its current location as well as creating a map. For what concerns this first point the tool RViz that, has been used as we'll see, it is able to create a map of the environment while the drone is moving, capturing the information of the obstacles given by the sensors (in our case LiDAR) and reading the IMU data that gives back the position of the drone.

Second feature, the drone must be able to measure and estimate its pose (position + orientation), in case of a real vehicle, the GPS is used to estimate its pose.

The amount of movement of the drone is measured by the odometry sensor. However, there is an error between the calculated distance with and the actual travel distance. Therefore, the inertial information from the IMU sensor can be used to reduce the error by compensating position and orientation error between the computed value and the actual value. Third, figuring out whether there are obstacles such as walls and objects requires sensors. Various types of sensors such as distance sensors and vision sensors are used. The distance sensors include laser-based distance sensors (LDS, LRF, LiDAR), ultrasonic sensors and infrared distance sensors. The vision sensors include stereo cameras, mono-cameras, omnidirectional cameras, and recently, RealSense, Kinect, Xtion, which are widely used as Depth cameras, to identify obstacles.

The last essential feature for navigation is to calculate and travel through the optimal path, the navigation algorithms, the global navigation algorithms and principally on the Obstacle Avoidance algorithms. The global path searching algorithms, as the A* and D* algorithms, always consider a starting point, a goal and a complete map that is periodically loaded; instead we analyse a different starting point and condition of the space. In this way we can do obstacle avoidance through 2d mapping and navigation.

LASER AND SONAR APPROACH:-

ABOUT LASER SCAN/LIDAR SENSOR-

The term 'LIDAR' is an acronym for Light Detection And Ranging or Laser Imaging, Detection, and Ranging. Lidar is sometimes called 3-D laser scanning, a special combination of laser scanning and 3-D scanning.It provides 360-degree high-resolution mapping, from short to long range.

Working of lidar is mainly based on the sending and receiving of light pulses between sender and the photodetector after reflection from the obstacle or object.Distance detection is one of the major functions of LIDAR. By measuring the travelling time of light pulse between the sender and photodetector after reflection of the target surface, the distance refers to $d = v * t/2$ (d, v, and t represent distance, light velocity and time between journey, respectively).

ABOUT SONAR/ULTRASOUND SENSOR-

Sound Navigation And Ranging (SONAR) sensing is an alternative for laser sensing in some fields, with similar working principles like RADAR. By providing a pulse of sound and listening to how long before the echo returns, such as with airplanes and underwater vessels, SONAR is able to provide high resolution and long range as well but presents better performance in different light conditions.

IMPLEMENTATION IN OUR MODEL-

- In our model, laser scan, sonar sensors are already present.

- Since, laser scan sensor has a larger field of view than the sonar sensor, we used the laser scan sensor to detect either an obstacle in the front or a corner in order to avoid the obstacle and travel further.

- We noticed that there was a /scan topic which was being published with the sensor readings, so we wrote a subscriber node which subscribed to the scan topic and displayed the sensor readings on the terminal.

- In this way, whenever the drone came close to an obstacle, the value kept on changing and we could set a value for threshold distance for implementing obstacle avoidance.

- For letting the drone fly at a constant altitude, we wrote a subscriber node which subscribes to a /sonar_height topic and using it we set the height at which we want our drone to fly.

PROS AND CONS OF USING ABOVE APPROACH:-

LASER SCAN/LIDAR-

PROS:-

- There are two methods to consider for choosing sensors for a particular task: active and passive. Generally speaking, the active method refers to sensors with laser sender and receiver, such as RADAR/LIDAR or ultrasound.Reliable obstacle detection systems must handle a huge amount of information and data, which is collected and processed in real-time. For this to happen, an “active” approach is employed and implemented, hence generally, laser scan,lidar,sonar sensors are chosen over cameras and infrared sensors for obstacle avoidance.
- Light laser has higher energy, higher frequency and shorter wavelength than radio waves, which gives a better reflection rate in non-metallic material. It can operate with a wider range of electromagnetic spectrum such as ultraviolet, visible and infrared regions.
- LIDAR has several advantages: accurate, wide field of range, long-distance range, irrelevant to different light conditions.
- LIDAR can also be used during night time because of its capabilities to work in low-visibility environments.

CONS:-

- The working of lidar sensors is limited by harsh environments like rainy or snowy weather and low-reflective targets.
- They are comparatively costlier and less flexible than cameras.

SONAR/ULTRASOUND-

PROS:-

- SONAR is able to provide high resolution and long range as well but presents better performance in different light conditions.
- For aircraft equipped with LIDAR, the sunlight might be too strong in the sky and cause problems when flying towards the sun, leading to an overexposure, in such a case a SONAR sensor may be useful.
- When considering underwater situations, for example, the ocean would absorb the light in red spectrum regions, which may cause worse accuracy if using laser sensing, in which case SONAR has proved to be useful.

CONS:-

- Because sonar is for near obstacle detection, it works well as a parking assistance system.
- Due to its short coverage range (<2 m) and poor angular resolution in terms of normal sonar sensing, sonar does not work well for obtaining information about the location and velocity of vehicles on the road and motorways.
- However, SONAR easily “suffers” disruptions in noisy environments. Many sources on the road, streets and highways generate noise: horns, engine vibrations or wind sound due to high relative velocity. In such noisy environments, laser sensing works better than SONAR and has more advantages
- If trying to enlarge the coverage range, the ping/pulse from the emitter can go very loud, which is harmful for citizens and is environmentally unfriendly.

WORKFLOW OF THE PROJECT:-

1) Simulate a ready-made model:

For the simulation of the drone, we have cloned a github repository that has a ready-made working model. Using this model, we have simulated our drone.

2) Read lidar sensor data with the help of a subscriber:

The second step of our project was to read lidar sensor data. We did this by importing the Range library from sensor_msgs.msg. It showed us the distance of our drone to any obstacle.

3) Perform goal to goal operation (autonomous flight):

Autonomous flight is the ability of a drone to perform aerial maneuvers without a human at the controls. Autonomy is when the drone performs self-flying actions without human input. The drone moves from one point to another on its own.

4) Perform obstacle avoidance:

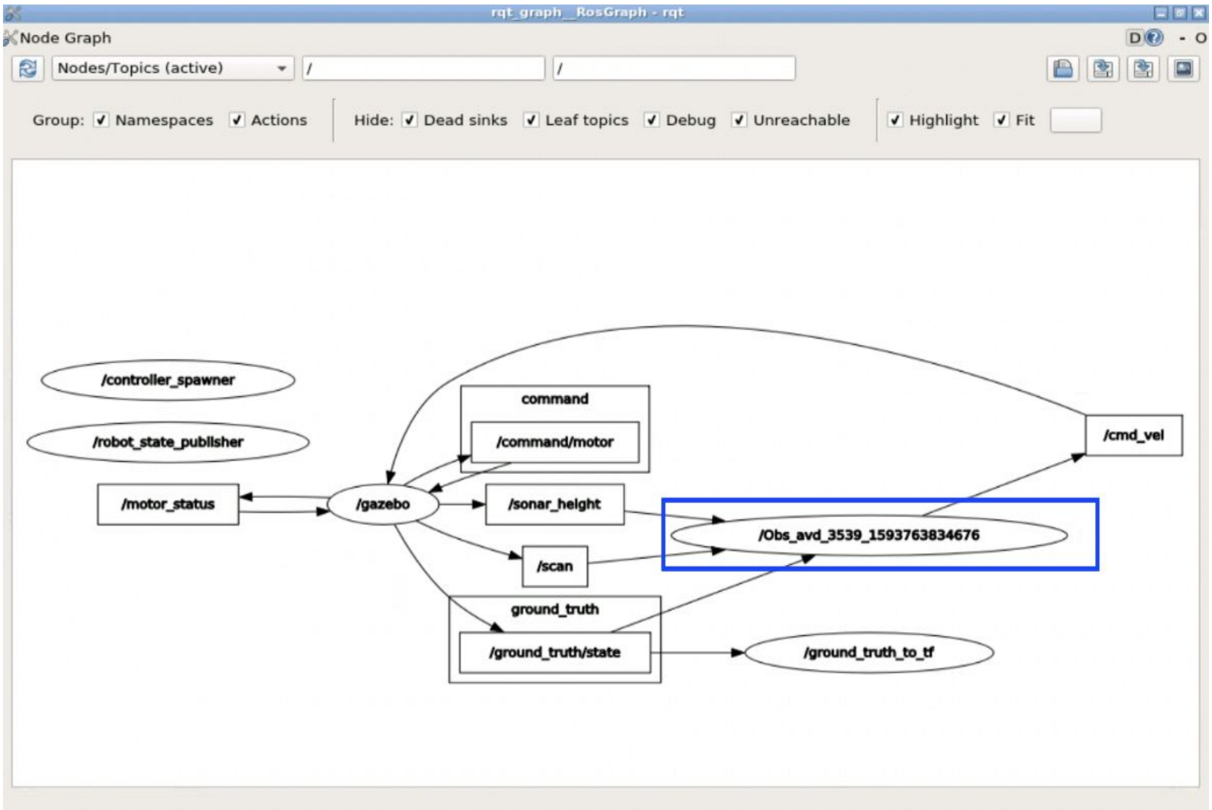
Autonomous Drone Collision Avoidance System is autonomously sensing obstacles and avoiding collisions during drone missions. We have performed obstacle avoidance by the drone.

5) Integrate it with each other:

The final step of our project was to integrate both goal to goal operation and obstacle avoidance so that the drone avoids obstacles and moves from one point to another autonomously!

PROCEDURE FOLLOWED:-

For our project , we have used the approach using Lidar and Sonar sensors.



1)Subscribers , Publishers and Topics :

The picture above shows all the active nodes and topics.

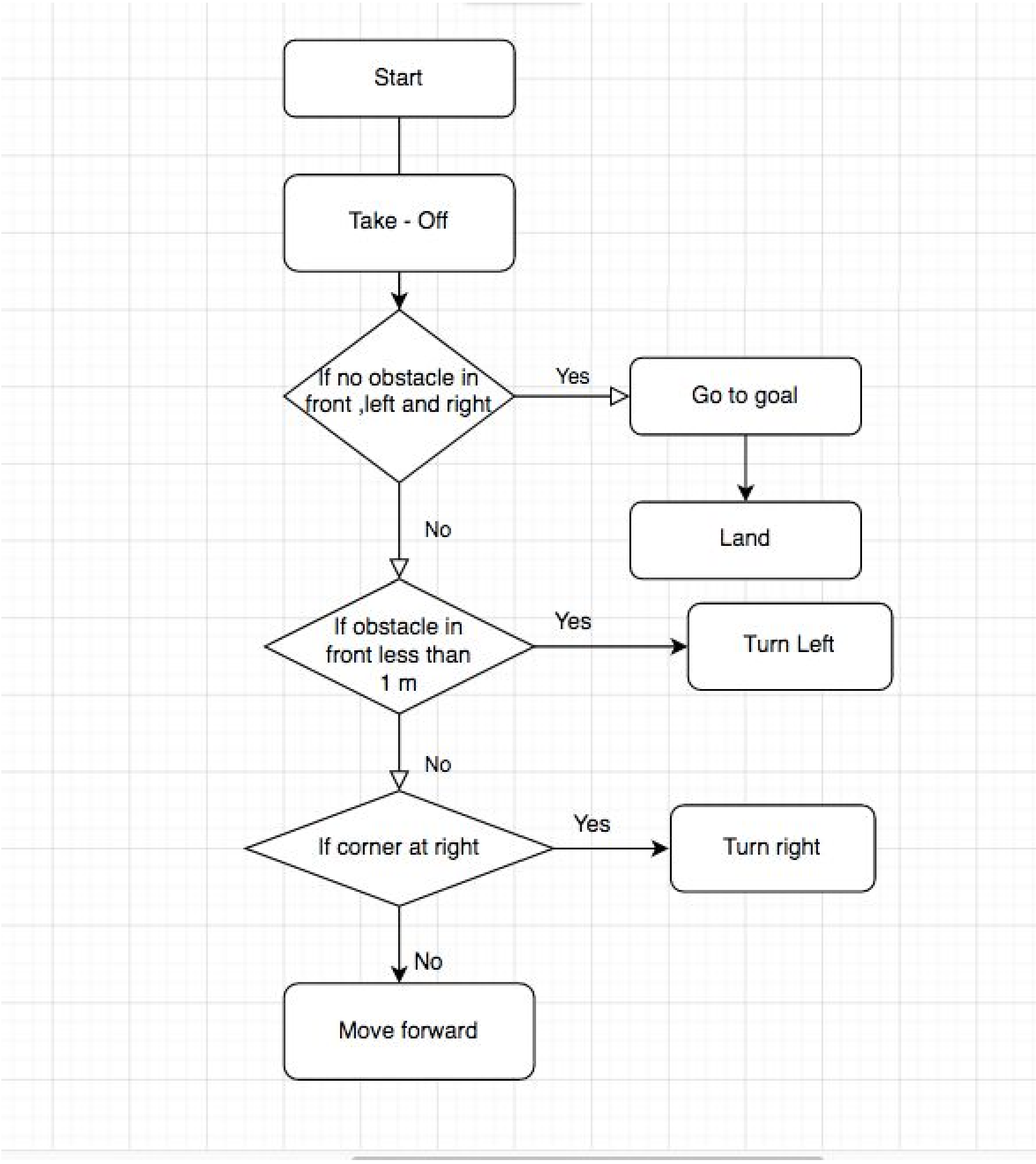
We have created the ‘obs_avd’ (obstcale_avoidance) node highlighted in blue.This node subscribes to ‘/sonar_height’, ‘/scan’ ,‘/ground_truth/state’ topics and publishes to ‘/cmd_vel’ topic as shown.The ‘/sonar_height’ topic is published by the sonar sensor and contains information regarding the height of the drone. It is used for take-off and landing of the drone.

The ‘/scan’ topic is published by the hokuyo_utm30lx lidar and helps in knowing the distance of the obstacles from the drone within a range of 0.1m to 30m.

The ‘/ground_truth/state’ topic is published by gazebo and uses messages of the type Odometry to communicate with its subscribers.This helps us in understanding the x and y coordinate of the drone and its roll , pitch and yaw values. We make use of this topic to find the current position of the drone which helps us to make the drone reach its goal. The ‘obs_avd’ node publishes messages on the ‘/cmd_vel’ topic to control the maneuvering of the drone. By publishing messages on this topic, we make the drone take-off , land , take a left or right turn and fly forwards.

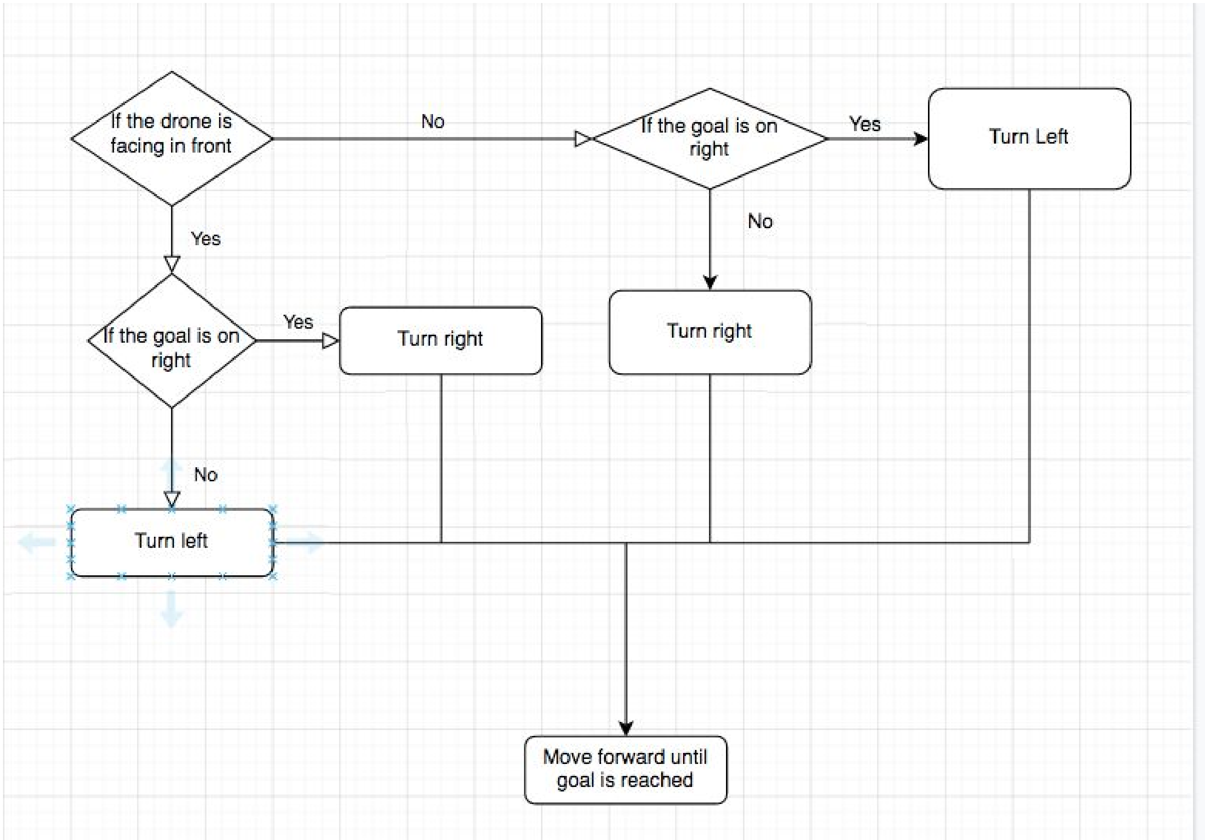
2) Algorithm for obstacle avoidance:

The following flowchart shows the algorithm we followed for obstacle avoidance.We have used the wall following method so that the wall always remains to the robot’s right.



3)Algorithm for goal to goal:

In this algorithm, we have taken the goal in the form of x and y coordinates. We have written two functions , one for making the drone reach the x-coordinate and the other for making it reach the y-coordinate.



The flowchart above shows the algorithm for moving to X-coordinate. A similar algorithm was written for moving towards y-coordinate.

TRIALS AND ERRORS PERFORMED:-

1) We used OpenCV which was not required.

2) We had to change our previous model as we were not able to add a laserscan plugin in it. This took us quite some time.

3) We tried using a sonar sensor but the lidar turned out to be better as it gives a 360 degree view.

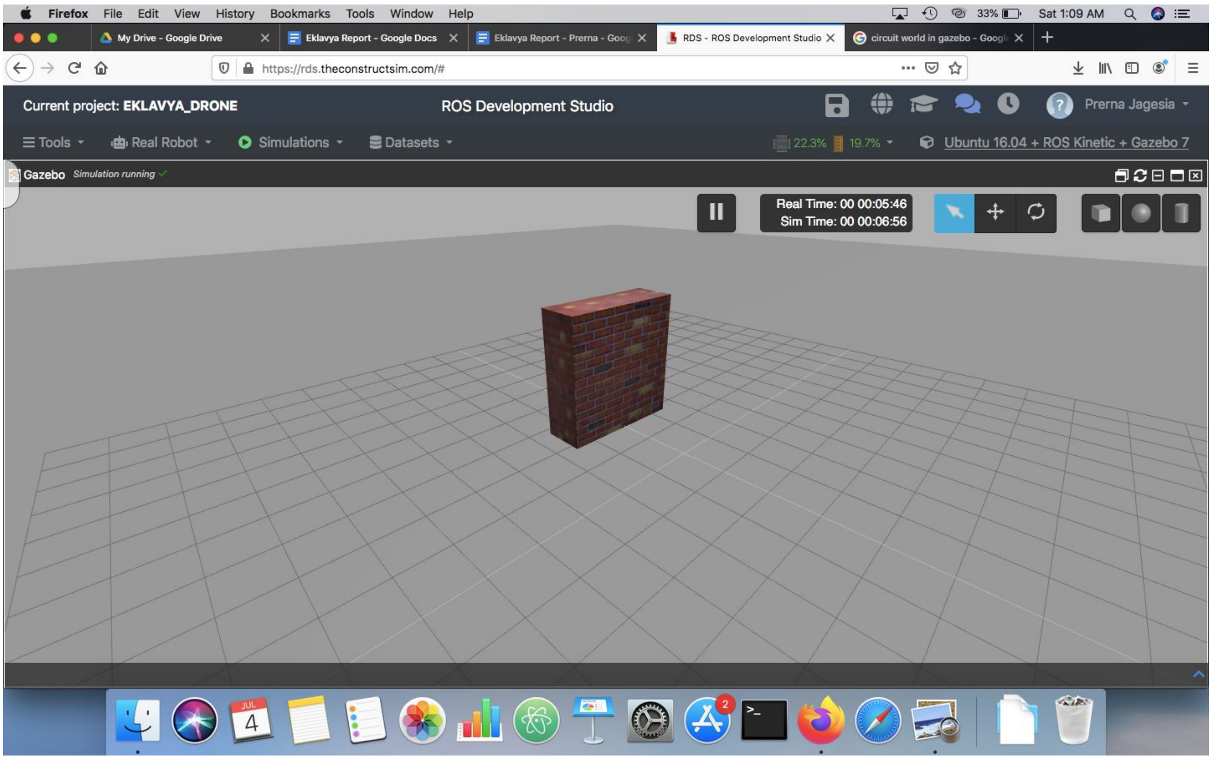
EFFICIENCY AND ACCURACY YOU ARE GETTING IN THE RESULTS:-

In our project , the drone will be able to avoid the obstacles in simple environments and work its way out of mazes as shown below. However for complex environments, our algorithm does not work well. It puts the drone in a circular loop in some cases. As we are not using a camera, the method we use to detect corners using a lidar is not so reliable.Sometimes, it fails to detect corners if the drone is made to fly with a very fast velocity.

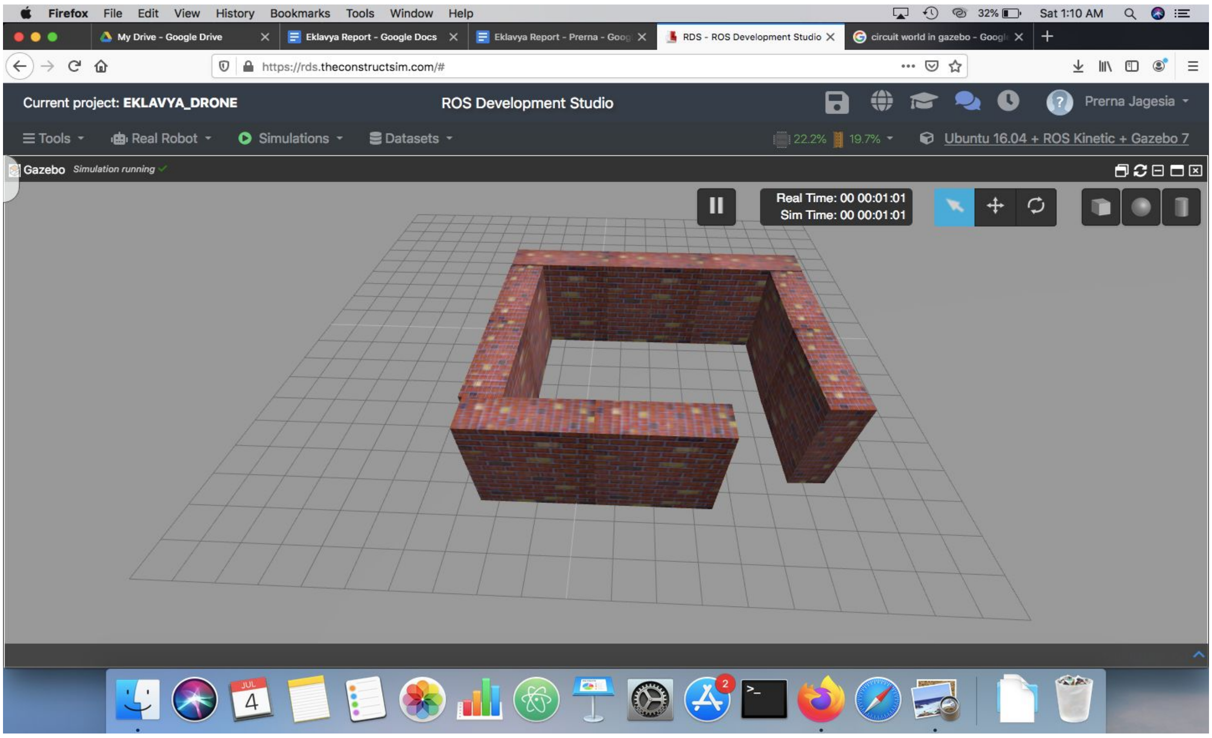
The drone will be able to reach its destination once it has avoided all the obstacles. This imposes another restriction. It cannot fly to places within the region of the obstacles. If it does so , it will collide with the obstacles.We need to find a better way to integrate goal-to-goal with obstacle avoidance. Also, the drone should be facing north during take-off, otherwise the algorithm will fail to work.

Environments in which it works:

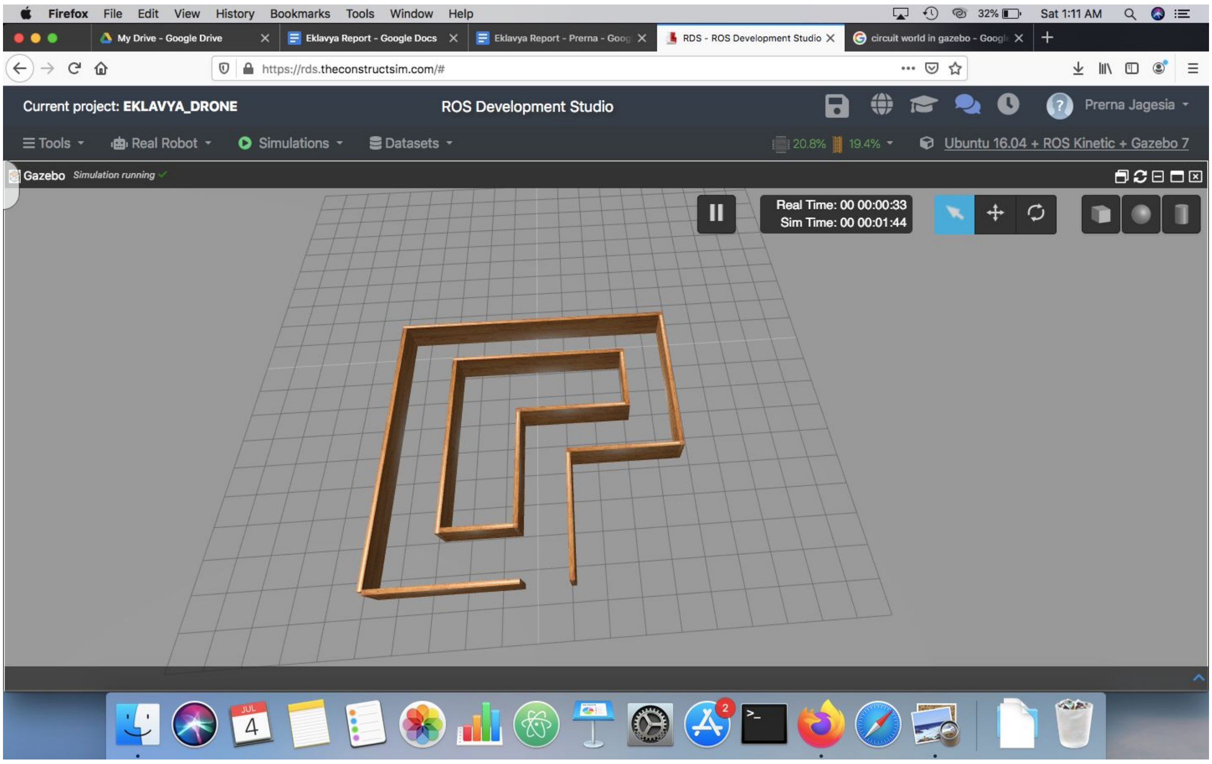
1)Empty world + wall:



2) Empty world + Four walls:



3) A Simple Maze:



WHAT DID WE ACHIEVE?

In this project, we have integrated goal to goal operation and obstacle avoidance so that the drone can fly from one point to another avoiding obstacles autonomously.

FUTURE ASPECTS OF THE PROJECT:-

Expanding upon the project , our first step is to integrate obstacle avoidance and the code which we have written for the drone to move from one destination to another within an environment of four walls . Our second step would be to implement obstacle detection code with obstacle avoidance as this would be a great aid for the drone to perform then future tasks like geography mapping , aerial photography without having to worry about obstacles in its way. Drones can be utilised for site surveys, monitoring, and mapping.
There are many ways in which obstacle detection algorithm can be developed in extension with obstacle avoidance. One such way is contour detection. Contour detection is based on detecting the outline of the object within the environment. It is preferred because it is based upon edge detection, which has been optimized in run time and complexity, therefore, allowing for near real-time run implementations. Further expanding on the advantages of contour detection, the edges and lines of the outlined obstacle can be used for avoidance applications, including but not limited to, optical flow, bug algorithms, and image segmentation for movement applications.
The detected contours of the objects present an easy interface for tracking points to be superimposed on the obstacle and tracked. These points would show the relative motion of the detected obstacle in relation to the quadrotor, allowing the quadrotor to react accordingly to its surroundings. From here, robust drone obstacle detection can be achieved.
Further expanding on avoidance, SLAM can be implemented so the quadrotor can create a map of the environment it is navigating. This brings forth the issues of computation power and memory limits; however, extra hardware can be used to achieve this goal. This means that the quadrotor system is no longer an 'off the shelf' system. If this path is to be chosen, a stereo camera can be mounted to the quadrotor for distance estimation. Supporting this, LiDAR can also be used, and the data acquired by the system can be processed and fused for a detection algorithm.

REFERENCES:-

This pdf was used to understand algorithm for obstacle avoidance-

1) https://link.springer.com/chapter/10.1007/978-3-319-62533-1_7

2) https://youtu.be/PyC4Vj3NUUY

This pdf was also used as reference for obstacle avoidance-

3) https://recipp.ipp.pt/bitstream/10400.22/14269/1/DM_ZubairKhan_2019_MEEC.pdf

4) **https://youtu.be/RFNNsDI2b6c**

Following 2 playlists were used for understanding basics of how drone works-

5) **https://www.youtube.com/playlist?list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y**

6) **https://www.youtube.com/playlist?list=PLgiealSjeVyx3t4N9GroE29SbVwhYrOtL**