Idea/Approach Details

Technology Bucket : Robotics and Drones
Category: Hardware
Company Name/ Ministry Name: Axiscades
Problem Code : ST1
Team Leader Name : Rajendra Singh
College Code : U-0878

Idea/Prototype/Solution

1) The solution for indoor drone navigation:

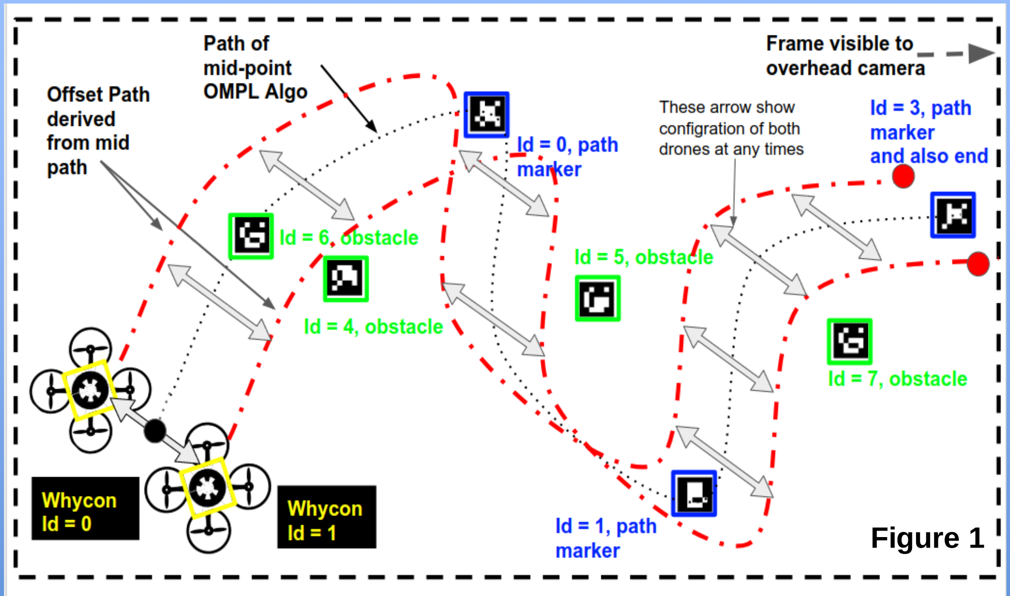


Figure 1

1.1) Introduction:

For indoor drone navigation, we're using markers for the path, obstacles and on a drone to locate them in a frame of an overhead camera which is watching this whole setup from the top. Consider figure 1, here

- A line of Aruco markers (blue boxes) is forming the path to be followed by drones.
- Aruco markers in the green box representing the obstacles
- Whycon markers (yellow box) are placed on the drones.

The overhead camera is able to locate each of this marker in its image frame and hence enabling both drones to navigate on the predefined path synchronously while avoiding obstacles

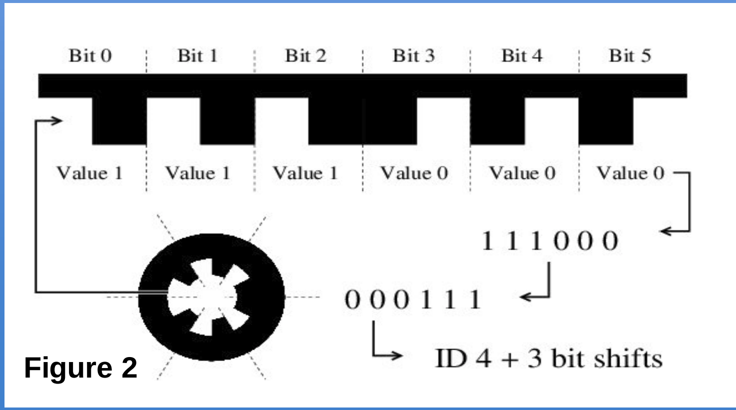


Figure 2

1.2)The position of drones:

There will be *Whycon markers* (see figure 1) on both of the drones. The entire arena is visible by an *overhead camera* so the Whycon markers can be detected. The detected Whycon markers will give us the positions and rotation of the drones in the frame of an overhead camera.

Figure 2. *This is Modified Whycon marker we're using to locate drones. Here, the inner circle of the Whycon marker encodes a binary string which is bit-shifted to match a Necklace code. Apart from identification, the number of bit-shifts allows us to identify the marker's rotation.*

In the case when a Whycon markers are not visible to the overhead camera, we can use IMU sensor data to locate the drone from a previously known point(where a marker was visible). We combine location coordinate of drone from IMU and overhead camera using Kalman filter to precisely locate drone.

1.3) Controlling drones:

We will use ROS (Robot Operating System) software to control the drones. The drones will be connected to the laptop via wifi. A master node running in the laptop will perform all the computation, path planning and will also control the drones by publishing to control topics. Both the drones will publish the sensor data to their corresponding topics and will subscribe to control topics for navigation. Navigation is possible in both ways - Automatic and Manual. We will use a PID algorithm for the same.

1.4) Path Planning:

Path planning of drones will be done in a software called V-REP (Virtual Robot Experimentation Platform) using Open Motion Planning Library (OMPL). V-REP will be used to create a simulation of the actual arena in a *virtual world*. Refer to Figure 1

The working of the *path planning* algorithm is as follows:

•The position of the destination point of drones and obstacles will be determined before running the algorithm.

•The drones will be kept at a distance which they have to maintain during the flight.

•Both drones will be considered as a single object to calculate the path, therefore, the path will be calculated for the *midpoint* of their initial position.

•Since only collidable objects are drones and the obstacles (NOT the midpoint), the OMPL algorithm will calculate the path appropriately for the midpoint.

•The drones will have to maintain the distance from the *midpoint path*, ultimately the drones will be maintaining the given distance from each other.

•Now that we have a path for the midpoint of the initial position of drones, we can divide the path in any number of points we want. Once we divide the path into way-points, one by one we will give points to the program which controls the drones, according to their distance.

•The next way point will be given to the program only if both of the drones have reached their previous way points. Hence, this way both drones are waiting for each other to reach their corresponding way-points thus forcing them to navigate synchronously while maintaining their relative distance.

•The waypoint will be considered as reached if the error in all the three axes is less than a particular given value.

This is the most **convenient way to navigate drone indoor** although this method is limited by having an overhead camera which may not be feasible everywhere. **Due to its simplicity, we used this method to validate our algorithm.**

1.5) Simulation & Node diagram:

Figure 3. *Simulation of navigation in VREP software. (for video click here.)*

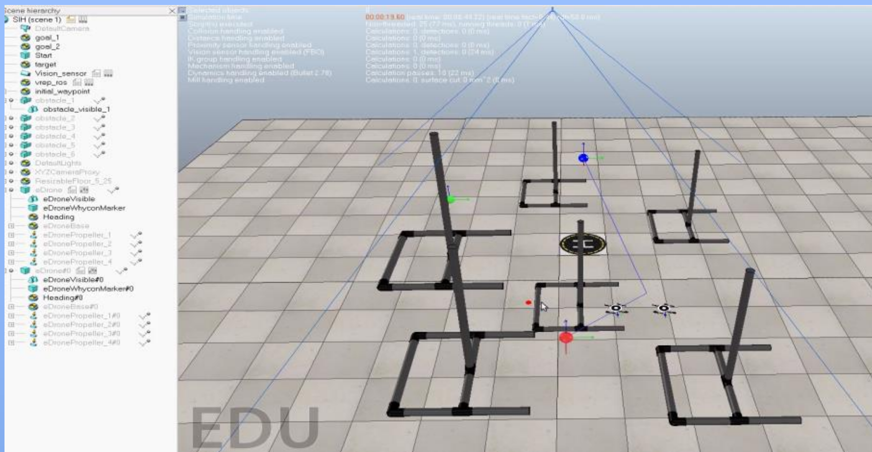
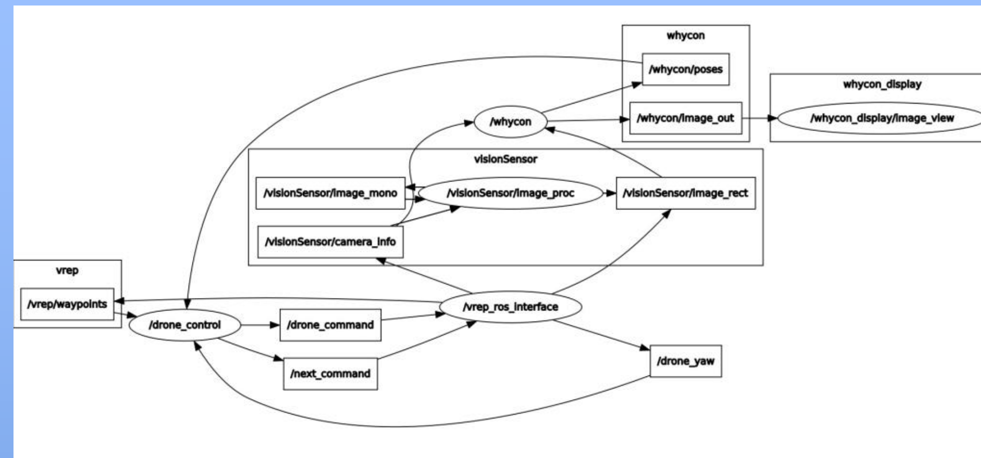


Figure 4. *Node diagram*



2) A solution for outdoor drone navigation:

2.1) Introduction:

For outdoor drone navigation, we're using GPS Module and front-facing cameras in both drones, from which we're able to locate both drones and obstacle w.r.t ground frame. Hence, able to navigate both drones between two positions synchronously, while avoiding obstacles.

2.2) The position of drones:

We can use a GPS coordinate of the drone to locate it in a ground frame and its camera to locate obstacles in the camera frame, further locate obstacle w.r.t. ground frame (figure 4). After that, we can use Camshift algorithm to dynamically plan the path to navigate around. But usually, GPS data is not much precise to navigate drone indoor. So, this method is the **best suitable for outdoor navigation** application only.

We will integrate INS (Inertial Navigation System) with GPS. An inertial navigation system (INS) uses the output of inertial sensors to estimate the drone's position, velocity, and attitude. A complete six-degree-of-freedom inertial sensor consists of 3-axis accelerometers and 3-axis gyros. This integration is necessary to locate both the drones w.r.t to each other and the world. For outdoor purpose or long distance, using a GPS is appropriate. But if the signal is lost, the coordinates will have high error. To avoid the error, we will estimate the drone's position, velocity, and attitude through INS. For indoor purposes, using INS will give high accuracy in terms of coordinates using Kalman filter to precisely locate drone w.r.t to ground frame.

IMU and GPS data are always synchronized within the resolution of one sampling interval. In this architecture, this is achieved by using the alarm functions provided by the RTOS kernel which are used to trigger module execution. Tuning the extended Kalman filter is known to improve filter convergence time and short term accuracy.

When the IMU is not collocated with the GPS antenna, the distance between the IMU and the GPS antenna results in different velocity and position measured by both sensors. This is known as the lever arm problem, and the Kalman filter must account for it to make correct state estimates. It can be resolved by transferring the GPS measurement to the location of the IMU by using the information of the aircraft attitude and the known lever arm vector in the body frame.

2.3) Controlling drones:

We will be using ROS (Robot Operating System) software to control the drones. The drones will be connected to the laptop wirelessly, through communication network(Figure 7) . Both the drone will be publish the sensor datas to their corresponding topics. Master node will be running in laptop and doing all the computation, path planning and will also be controlling the drone by publish to control topics. Both drone will be subscribing to these control topics for their navigation.

2.4) Navigation:

Waypoint GPS Navigation allows a drone to fly on its own with it's flying destination or points pre-planned and configured into the drone remote control navigational software. Waypoints to be selected will be of the midpoint path.

- Store all the waypoint coordinates.
- Check the next waypoint. Depending on UAVs position, fly straight or turn with a constant angle(depending on path given and presence of obstacles). When it gets there, perform the same process toward the next destination with the following algorithm:(Figure-5)

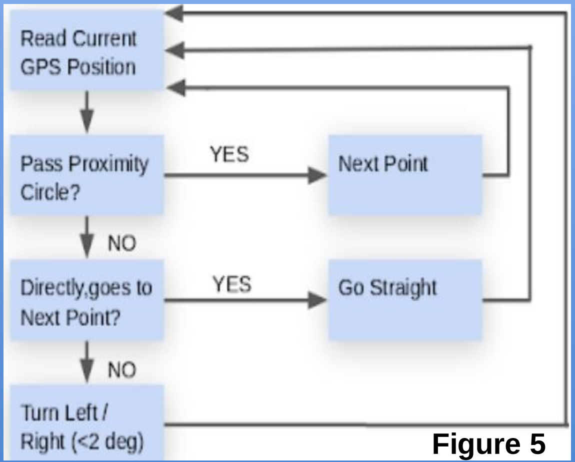


Figure 5

Step A.) Read current GPS position from the GPS receiver for both drones.

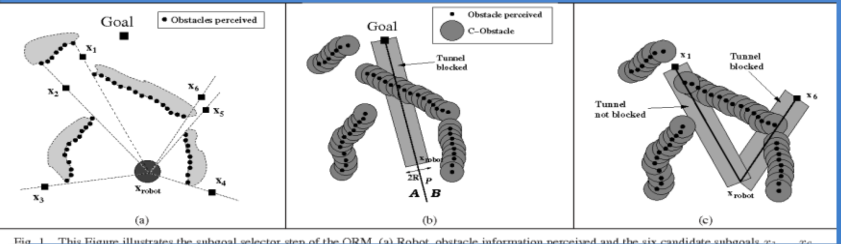
Step B.) If both UAV passes the proximate circle, then continue to the next point and go to step A. If not, go to step C.

Step C.) If UAV is on the direction to the next point, go straight and go to step A. If not, Turn left or right a given degree(depending on an obstacle) and go to step A.

- Encounter Obstacle:

To avoid obstacles in the flight, we will use the obstacle-restriction method like VFH (Vector Field Histogram) algorithm, to avoid an obstacle and move in the available window (Figure 6). Since we consider both drones altogether making it into a single object , where only drones are collidable. So the path will be planned for the midpoint of this combined object. Using which we can drive the path of individual drone similar as in case of indoor navigation. Also since the relative distance between this driven path will be constant (as driven from the same midpoint path). Thereby, both drones will be able to maintain their same relative distance and attitude too. For efficient obstacle(3D) avoidance SLAM(Simultaneous Localization and Mapping) algorithm like Hector-SLAM, LSD-SLAM will be used.

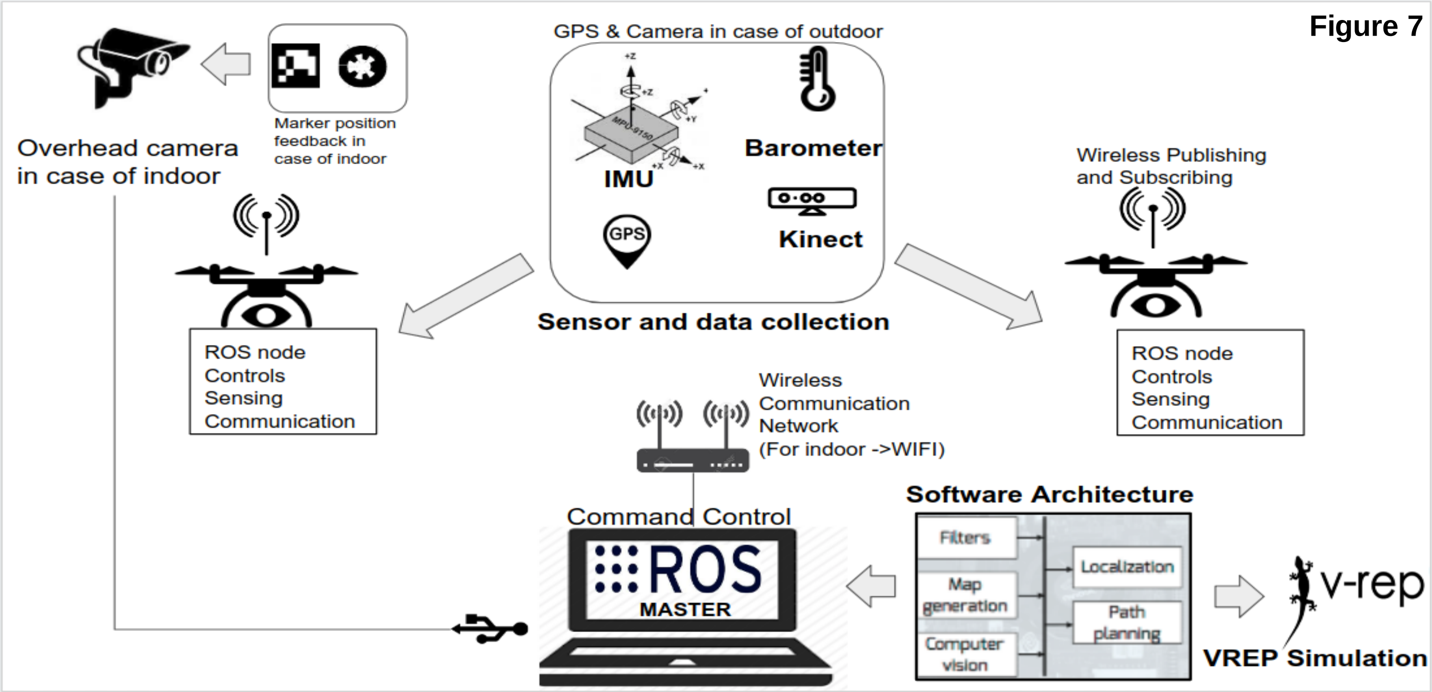
Figure 6



2.5) Transfer of Information:

For both the drones, we will make dynamic nodes to share information using ROS. Information like GPS coordinates, estimated location from INS, estimated path swept using the camera, attitude will be exchanged over nodes, in order to calculate relative distance, relative coordinates and relative attitude. And these values will also be exchanged.

Tech Stack:



Use Cases:

- To balance an extended object while lifting, delivery.
- To create a wide-angle view image by stitching two images taken from parallel drones.
- Remote sensing and exploration.

Show stopper / Limitations:

- Path planning using markers and an overhead camera is only an indoor solution.
- Image processing would be needed for obstacle detection in GPS. The obstacles will only be detected when they are in a certain range of the camera.
- The range of Wifi limits how far the drone can be sent.
- GPS has a low update rate (5 Hz).
- The algorithm in case of GPS will take more time to execute due to the secondary drone calculating its path from the path of the primary drone.

Dependencies:

- A Linux distro preferably Ubuntu 16.04 is required to support ROS and VREP.
- Markers for localization & overhead camera for marker detection are needed.
- INS must be attached to the drones.
- GPS, Wifi and camera modules must be installed on the drone.