Unmanned Aerial Vehicle of Team IFOR for the International Aerial Robotics Competition 2017

Saptadeep Debnath, Anudeepsekhar Bolimera, Srinath H. Rao,
Ayush Shirsat, Parv Khandelwal, Sarthak Bhooshan, Sujith Sizon, Bhuvesh Gaind, Shaleen Bengani, Navya Zaveri, Sachin Samuel, V. Kalaichelvi

BITS Pilani – Dubai Campus

ABSTRACT

This paper describes IFOR (Intelligent Flying Object for Reconnaissance) an autonomous aerial vehicle that has been developed by Team IFOR at BITS Pilani, Dubai Campus. The vehicle being discussed in this paper is able to localize and navigate in an indoor GPS denied sterile environment. Downward facing Optical Flow sensor Px4Flow is used to localize and stabilize the attitude (pitch, roll and yaw) and altitude of the vehicle using PID controllers. Ground robot detection is carried out by two inclined forward facing cameras and uses a LIDAR to avoid obstacle ground robots by setting the area near the paths of obstacle ground robots as hazardous area.

1 INTRODUCTION

1.1 Statement of Problem

The Mission 7 of IARC, is to develop an unmanned autonomous aerial vehicle, whose aim is to herd 10 ground robots across a green line in a 20x20 m arena, through physical interaction via tapping or landing in front of it. The UAV must carry out this task whilst detecting and avoiding the four moving vertical obstacles. The mission arena is located indoors, thus the UAV cannot rely on the external localization techniques such as GPS or SLAM algorithm.

1.2 Conceptual Solution

Our proposed method concerns itself with the use of a tested UAV which will be carrying out all the required operations that will help us achieve the mission statement with ease. Utilizing cameras in both horizontal and vertical axes, transmission of a live video feed to an Off-board station will be performed over Wi-Fi where the image processing procedures are applied. Ellipse Detection is used to identify the ground robots. The Onboard computer will be performing the primitive operations such as flight control, position control, and coordination of all the various on-board sensors like Px4Flow, LIDAR, etc. The Px4Flow is primarily used for drift control and position hold. The LIDAR, on the other hand, sends pulsed lasers to measure ranges of the obstacles which thus helps in obstacle avoidance. The Path planning software will assign prioritized instructions to ground robots according to the proximity of each robot to the target. All of the operations discussed, will ensure the drones to navigate the ground robots across the target within the allotted time limit. The image processing techniques and overall system architecture is discussed in detail in this paper.

1.3 Yearly Milestones

Team IFOR is active in this competition since the 2011 version, in which it secured first place at the American venue. The preparation for this particular mission began in June, 2014. During the years 2014-15, the team worked to develop and modify the old UAV (Asctec Pelican) to suit Mission 7a. In the year 2015-16 the platform was changed to UFO type quadrotor. The team is divided in to small groups which handles the various sections such as Object Identification and Tracking, Flight Control Loop, Sensor suite, Flight Electronics and Structures. The team also possesses a 3D printer, which has been involved in designing various custom parts for the UAV. The UAV used for the mission has been customized including the sensors, motors, propellers, ESCs etc., which will be used for simulation and software analysis purposes, to provide optimal results during the final mission.

2 AIR VEHICLE

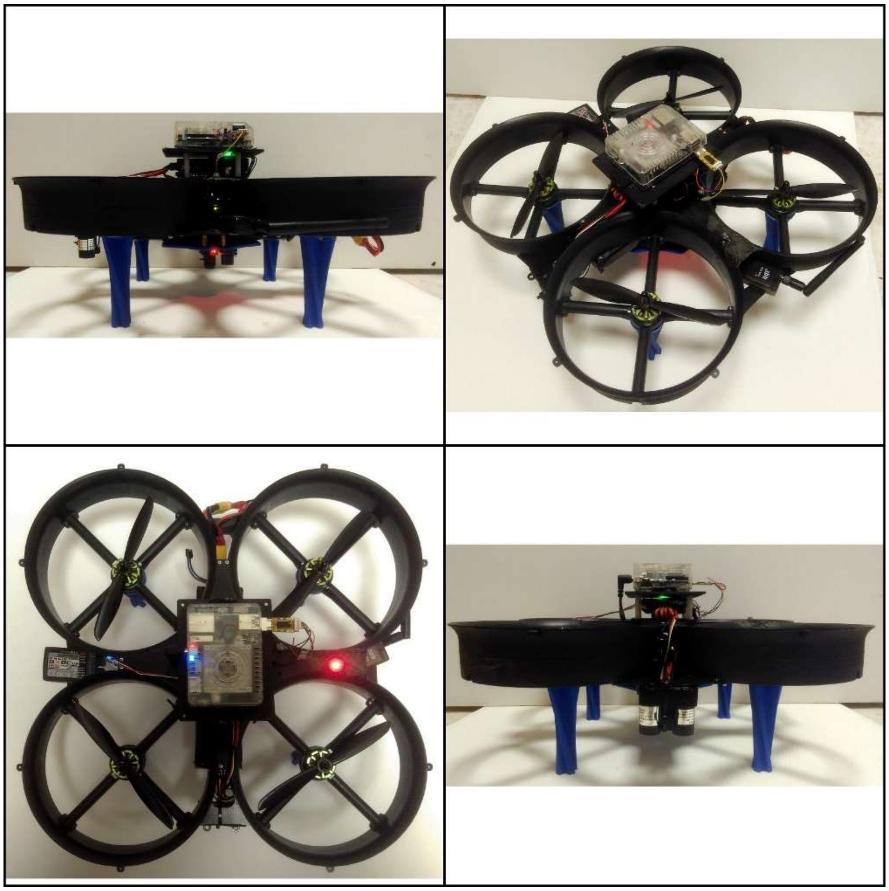

Figure 1. Side view, isometric view, front view and top view (from top left in clockwise)

2.1 Propulsion and Lift Systems

The most important part of the quadcopters design is its propulsion system as it is responsible for more than 80% of the total power draw. Thus the propulsion system should be efficient and appropriate to the mass of the UAV in order to achieve longer flight time. Figure 1 demonstrates the air vehicle used by Team IFOR. Our choice for the motors were the Multistar 2206-2150KV V2s for their small size, high speed and low power consumption. Each motor is paired with a

12A ESC. The motor's lightweight and explosive acceleration made it an optimal choice for our propulsion system design. For the propellers our choice was the 6030 propellers by Gemfan. The 6 inch propellers with a 3-inch pitch paired with the 2150KV motor and a ducted quadcopter frame provided a thrust to weight ratio of 1.8 and a very stable flight. The propulsion system is powered by 2 11.1-volt Lithium Polymer batteries with a discharge rate of 25C and a capacity of 5000mAh each. The quadcopter with the above mentioned configuration was able to achieve quick acceleration with a stable hover time of more than 12 min in our testing.

2.2 Guidance and Navigation Control

2.2.1 Stability Augmentation System

The task of controlling unmanned quadcopters is rather complicated in the presence of parametric uncertainties and measurement noises. It is essential that the flight control system of an unmanned quadcopter should be endowed with well-suited automatic capabilities to carry out complex flight missions. Our vehicle uses an off the shelf flight controller, i.e. the Pixhawk flight controller an open-source hardware project by 3D Robotics uses a wide sensor suite to achieve the desired attitude and altitude. The Pixhawk uses 2 level PID controller to control the motor speeds to achieve the desired attitude (i.e. the roll, pitch and yaw) and altitude, and removes the hassle of dealing with the dynamic nonlinear complexities of the quadrotor system allowing us to focus on the autonomy of the quadrotor itself.

2.2.2 Navigation

The video feed from the camera is used to update the locations of the ground robots and obstacles every five seconds. The intermediate locations are probabilistically estimated using our machine learning algorithms. Using the position estimates, a graph consisting of the ground robots and obstacles is created and a cost is assigned to each ground robot. The cost is a function of the ground robot's distance from the air vehicle, distance from the boundary lines and proximity to obstacles. A cost is also assigned to each square in the arena. Squares close to, or in the trajectory of the dynamic obstacles are given very high costs and are avoided during path planning, while squares containing the ground robots have lower costs. The squares with low costs act as waypoints. A modified version of Dijkstra's algorithm is run on the graph to figure out a straight line path along the waypoints to the ground robot to target.

2.2.3 Control Architecture

The control architecture of the system implemented, involves a two tier loop. One of them is executed on the Pixhawk while the other one is executed on the ROS framework which runs on the onboard computer. The set points (x, y, z, roll, pitch and yaw) generated by the path planning node goes through a feedback attitude controller which reduces the error. This value is then provided on to the outer control loop which runs on the Pixhawk flight controller. The outer loop attitude controller calculates the resultant torque values required to balance out the motion of the vehicle. Extended Kalman Filter is used as the estimator to filter out the calculated values and to reduce the error in the measurements.

2.3 Flight Termination System

The quadcopter is equipped with a manually operated kill switch which can stop the quadcopter when it enters an irresponsive or harmful state. The kill switch is connected between the battery and power distribution board. We have an independent receiver for the kill- switch from the Onboard computer. When we trigger the kill switch a pulse is generated and it stops the power supply between the motors and the battery. A channel on 9ch remote control is given for manual override of the quadcopter in case of emergency.

3 PAYLOAD

3.1 Sensor Suite

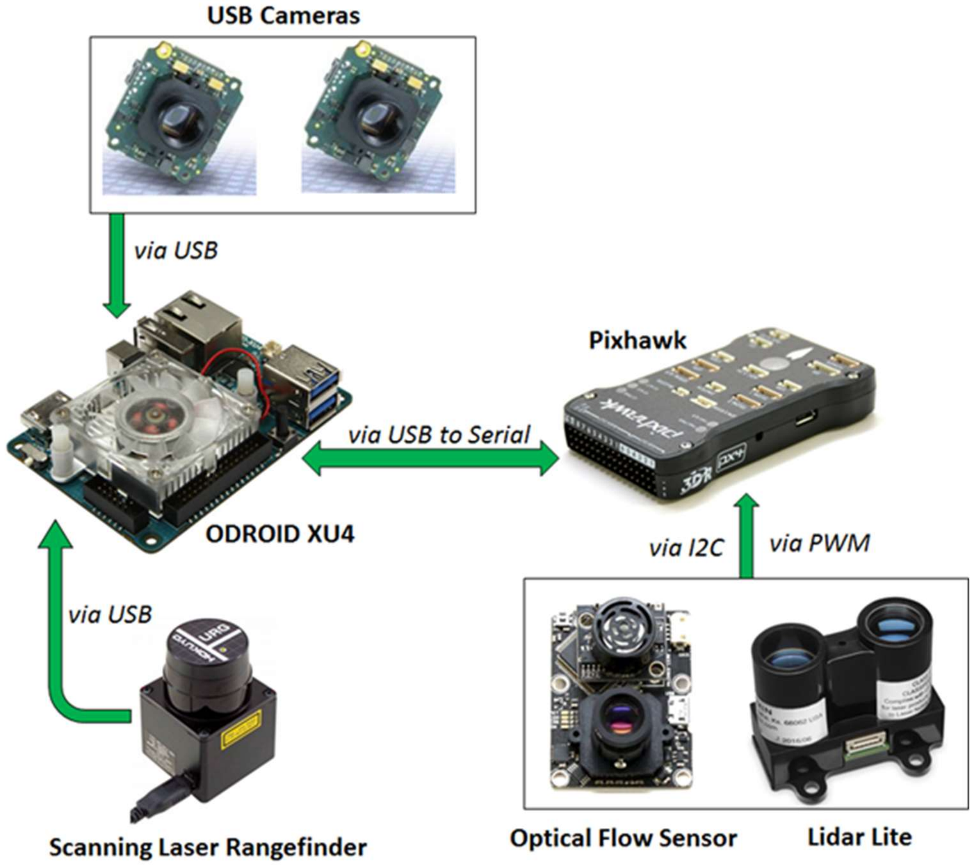

Figure 2. Sensor suite architecture

3.1.1 GNC Sensors

- Hokuyo URG-04LG-UG01 Laser Rangefinder-: This laser rangefinder assists the drone in avoiding obstacles, mounted horizontally, it has a range of of 20mm to 5600mm (1mm resolution) and a 270° field of view with 0.25° angular resolution. It also has a scan rate of 40 scans/sec. These specifications allow us to detect obstacle robots with ease and make necessary amendments to our algorithm to avoid them. Its lightweight and low power consumption property is an added advantage.
- PX4FLOW Optical Flow Sensor: The PX4FLOW Optical flow sensor will be mounted horizontally. Using its downward facing camera, it calculates optical flow and determines aircraft ground velocity. It can work in low light conditions without the need for any form of illumination. This property along with its ability to be reprogrammed made it an essential choice.
- LIDAR Lite V3: This LIDAR allows the UAV to calculate the optical distance thus

providing the distance from the ground at any point in time. The laser has a range of 0-40m, and provides an accuracy of +/- 2.5cm at distances greater than 1m, falling well within our requirements for the competition. The LIDAR lite is compact and weighs less allowing us to reduce our payload. The low power consumption (<140mA) is also an added to increase our flight time.

- Pixhawk Autopilot: The autopilot is responsible for calculating roll, pitch and yaw angles as well as the rates. These values are then processed in the control loops present inherently in the Pixhawk.

3.1.2 Mission Sensors

Cameras: Our UAV utilizes two mvBlueFOX-MLC cameras. The cameras have an onboard memory up to 8 Megapixels with a resolution of 752x480 with a max frame rate of 95Hz. The driver also reduces the PC load during preprocessing. These images are then processed to identify ground robots and take necessary actions.

Target Identification-
The vehicle is equipped with two cameras facing the ground at an optimal angle so that it covers the maximum area possible. The raw image input from the cameras are stitched together and the resulting image is used for ground robot surveillance. After surveillance ground robots will be detected with the help of Object Detection Algorithm as shown in the figure 3. As the ground robot would look like an ellipse when viewed from an angle, so our algorithm uses ellipse identification to detect the ground robot.

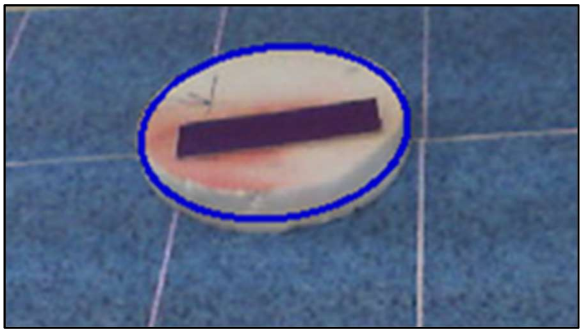

Figure 3. Object detection test on a sample ground robot

After a series of performed tests using the Object Detection Algorithm, the results achieved, as shown in the figure, were fairly accurate given the available resources.

The algorithm implemented returns various parameters which would be used for the probabilistic estimation of the future position of the ground robot and on the basis of this we prioritize ground robots. Those nearer to green line will get a higher priority and after detection, our soul motive is to navigate the ground robot across the green line.

After receiving the ground robot priority order from the object detection node, the position of the ground robot with highest priority is sent to the path planning node via ROS. The path planning for the UAV is done on an offboard computer because of its computational requirements as it utilizes multithreading to factor in a lot of information like obstacles positions, local position/

velocity of the UAV, odometry information, proximity to the boundary lines etc. and returns an optimized path to the particular ground robot. The UAV then enters ground robot follow and trigger mode. Based on various inputs, the target's movement with respect to the finish line is predicted using the density map generated from ground robot simulation. Mission coordinates are generated in such a way that it lands in front of the ground robot and makes it move in the favorable direction. These coordinates are then sent to the UAV's navigation node. The UAV stays locked on to the particular ground robot till it crosses the green line then continues to take new set of priority order for ground robots.

Threat Avoidance-
The quadrotor will fly near the maximum allowed height of 3m for most of the flight in order to avoid obstacles and to have a larger field of view. When flying at heights less than 2m, the LIDAR range finder is used to look for nearby obstacles. The readings from the LIDAR are then fed onto an obstacle avoidance algorithm which also runs on the main ROS framework.

3.2 Communication

Our vehicle communicates with a base computer in order to deliver telemetry data. The communication system consists of a 2.4GHz Wi-Fi Channel for transmitting data and video. A connection between the ODROID and the Pixhawk is established using a FTDI cable and a mavros protocol. A SSH (Secure Socket Shell) is also set up on the ODROID for transmitting data securely over the unsecure Wi-Fi network. The ODROID is used as a companion computer, which handles a lower level of processing as compared to our ground station. Finally, the Pixhawk is connected to an RC transmitter via a 2.4GHZ radio channel.

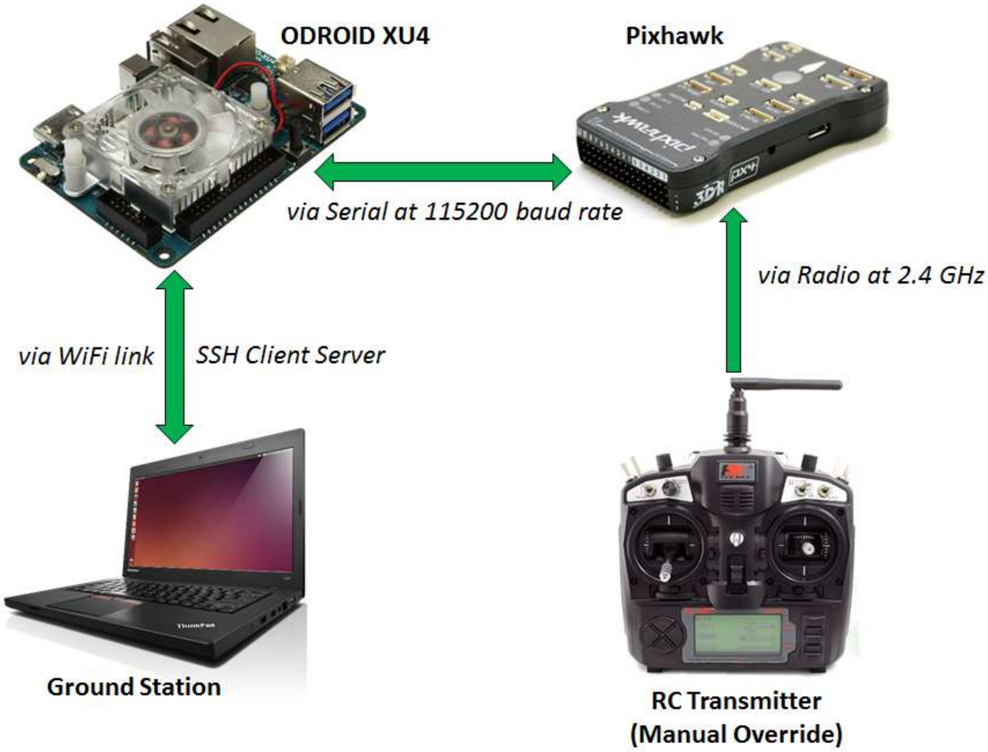

Figure 4. Overall communication setup

3.3 Power Management System

The quadcopter is powered by two 3 cell LiPo battery in parallel which has a nominal voltage of 11.1V and a capacity of 5000mAh. It also has a minimal discharge rate of 25C and maximum burst rate of 50C. The flight time is of approximately 12 minutes which is appropriate for the competition standards. Battery is connected to the ODROID via a UBEC voltage regulator which provides 5V supply and battery voltage is monitored continuously for flight safety.

4 OPERATION

The vehicle predominantly operates autonomously. However, a man machine interface has been incorporated for testing, ensuring safety and to monitor the vehicle status.

4.1 Flight Preparations

- Check physical condition of the vehicle; all components should be in their designated place and the propellers must be securely tightened
- Check the status of the battery
- Turn on the safety switch and enable the Kill switch.
- Establish SSH connection between offboard station and onboard computer over Wi-Fi
- Arm the vehicle
- Launch mission ROS package

4.2 Man/Machine Interface

During experimental mission executions, the Man Machine Interface is used mainly for monitoring purposes. It has a terminal console for launching the ROS nodes and a visualization of the image processing backend running on the system. The terminal console shows a summarized status of each modules, the current action and its execution status, local location, target location and battery status. Extra terminals show a visual plot from ROS Rviz showing various topics and nodes and how they are connected. The operator only interacts with the man machine interface to start the mission by executing a single launch file.

During testing, the MMI has a specifically tailored set of launch files for testing various components of the quad individually. It allows the operator to do experimental tests focused on specific behaviors or parts of the architecture.

Six channel RC controllers are used to switch between manual flight mode and autonomous mode and also acts as a kill switch by cutting off the power to propulsion system. All data is logged for future review and debugging.

5 RISK REDUCTION

5.1 Shock/Vibration Isolation

The Pixhawk is mounted on shock absorbing material, to reduce the vibrations being generated due to the propulsion system. In contrast to an earlier stage of testing, the vibrations were also reduced by the use of low pitch propellers, i.e. a change from 0645 to 0630 type propellers.

In addition to this, custom made landing gears are 3D printed and fitted on the vehicle. These landing gears are designed to absorb maximum amount of shocks from abrupt landings.

5.2 EMI/RFI Solutions

The flight controller which is effected by the EMI is mounted on the top of frame away from the ESCs and the motors. There is no radio frequency interference as there is no video transmitter on the system. Also there are two receivers with different frequencies for motors and kill switch, which don't interfere with each other.

5.3 Safety

The vehicle is equipped with some of the safety measures like the guards around the propellers and the emergency kill switch.

The only part which should be protected from the environment are propellers. Due to its UFO design there are cylinder shaped guards (figure 5) around the propellers of appropriate thickness to protect the quadcopter from damage caused by obstacles, as well as the surrounding bystanders from the propellers.

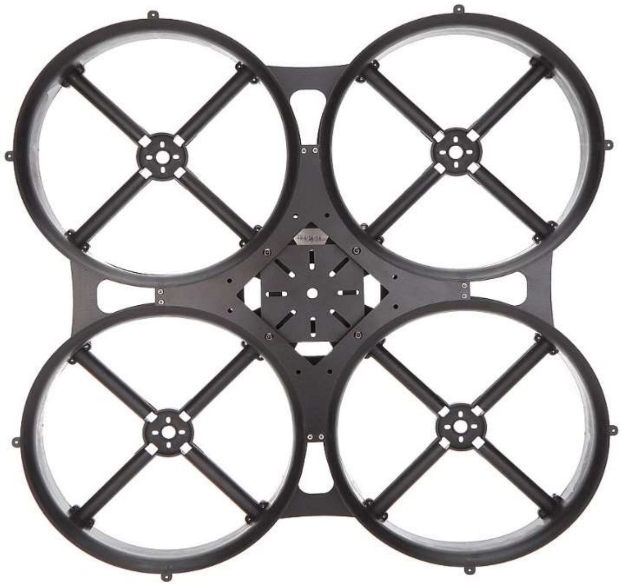

Figure 5. UFO frame as the base of the Quadcopter

5.4 Modelling and Simulation

The simulation of the ground-robots and obstacle-robots was done with Gazebo Robotics simulator. Gazebo – a full physical simulator – allows for the simulation of robots in complex indoor and outdoor environments. Gazebo's physics engine was deployed to design the robots and account for variables such as collision factor, friction, etc.

To simulate the movement of the robots, we used ROS (Robot Operating System). This acted as an interface, allowing us to mimic real-life obstacle and ground robots on Gazebo through ROS packages. In addition to this, ROS also made it possible to test out the random behavior of the robots, which proved useful when we tried to develop strategies for the Mission. The code for the movement of the both obstacle and ground robots was written in C++, while the code for designing the robots was written in XML.

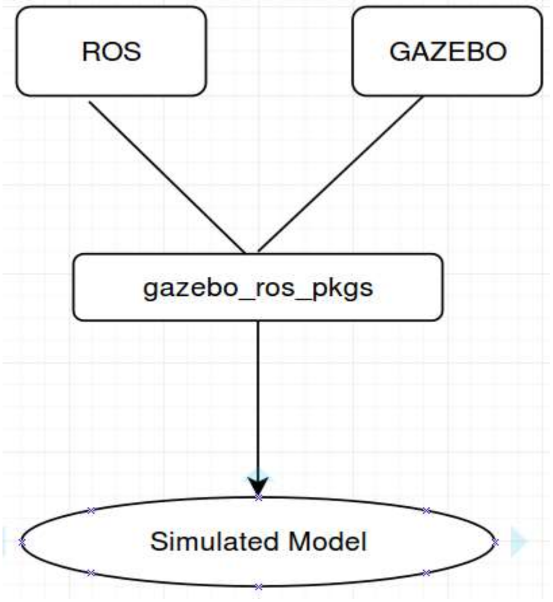

Figure 6. Schematic flowchart showing dependencies used in making the simulation

Building on the simulation, we designed a neural network to help accomplish the mission. The neural network was made to find the relative probability of a given robot being in specific regions of the arena. The data thus acquired optimized the quad-copter's working so that it would (selectively) “herd” only those ground-robots closest to the edge of the arena, and perform other tasks just as efficiently.

5.5 Testing

Testing is accomplished before starting the mission, so that the mission can be achieved safely and according to the regime. Testing is categorized into four parts- Calibration, Free flight testing, Gazebo Testing and Mission Testing.

5.5.1 Calibration

Calibration of the drone includes calibrating the IMU sensor, the motors and the speed controllers. This is done in order to ensure its accuracy. Calibration of the drone is done using the QGroundControl, which is a software used for the former. The QGroundControl software automatically collects relevant data for each motor, speed-controller and propeller combination with the aid of the flight controller, the 3DR Pixhawk.

5.5.2 Free flight test

After the calibration is completed, the UAV is tested for a free flight. Though the motor/propeller/speed-controller triad has been attested, the former is performed because the control-loop of the 3DR controller still needs to be tested and its PID gains need to be tuned. For safety purposes, ropes are attached to the vehicle. Initially, the height is manually set using a joystick or the controller. The vehicle is raised roughly 20 cm off the ground to verify 3DR functionality and to fine-tune x, y, and yaw stability in the outer loop. Once the stability is achieved, the vehicle is slowly raised to an operating altitude of 1 m. Slight adjustments are made to account for leaving the ground effect zone. Once this stability is achieved the PID gains for height control are tuned. Following this, x, y, yaw, and height set points are sent to the UAV from the ground station. These set points are altered by moving the joystick. Before autonomous movement is attempted, movement in each direction is tested. After the stability of the outer control loops has been checked, preprogrammed autonomous movement is tested. Once the proper vehicle response is checked, the surrounding environment is located and mapped with the help of onboard sensors. Next, the actions of the quad-rotor are checked and examined with the help of a GUI. The quad-rotor performs these actions if it is in autonomous mode.

5.5.3 Gazebo Testing

The mission was simulated with Gazebo, a robot simulating software. The movement of the UAV, ground-robots and target-robots was tested out with Gazebo. In addition to this, the simulated model allowed for the virtual deployment of real-world variables such as: Lidar, cameras, Pixhawk flight controller unit and px4flow.

5.5.4 Mission Testing

An environment similar to that of the Competition Arena is built, on which we implemented all the strategies and navigation techniques that we formulated, as mentioned in the former sections of the paper, on a single ground robot. Finally, the exploration functions are enabled and the vehicle is ready to fly the mission.

6 CONCLUSION

In this report by Team IFOR, we present a very unique approach for tackling the Mission 7a. It is done by integrating simulation results and real life consequences, and running them through a machine learning algorithm providing certain parameters which gives optimal results when fed into the vehicles operating system. Team IFOR's vehicle is currently in the autonomous testing phase, where it can detect the target and follow the same.

7 ACKNOWLEDGEMENT

We are indebted to BITS Pilani Dubai Campus for supporting our logistic requirements. We would like to thank our director Dr. R. N. Saha, who provided positive insights, that greatly emphasized the team dynamics. We would like to express our heartfelt gratitude to our team's advisor Dr. V. Kalaichelvi for her extensive support towards us to help us prepare for this mission.

8 REFERENCES

[1] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," Control Systems Technology, IEEE Transactions on, vol. 14, no. 3, pp. 562-571, may 2006

[2] J. E. Gomez-Balderas G. Flores L. R. Garcia Carrillo R. Lozano, tracking a Ground Moving Target with a Quadrotor Using Switching Control, J Intel Robot Syst (2013) 70:65-78, DOI 10. 1007/s10846-012-9747-9.

[3] Lorenz Meier et al., "A System for Autonomous Flight using Onboard Computer Vision", Proc. of The IEEE International Conference on Robotics and Automation (ICRA), 2011.

[4] Official Rules for the International Aerial Robotics Competition - Mission 7, May 2016, http://aerialroboticscompetition.org/downloads/mission7rules_051016.pdf.

[5] PX4 Development Guide, Pixhawk, n.d., http://dev.px4.io/index.html.