DRDO S.A.S.E'S UAV Fleet Challenge

8th Inter-IIT Tech Meet

IIT ROORKEE

Mid Evaluation Report

Presented by: Indian Institute of Technology, Mandi

TEAM :
Ritwik Saha(Leader)
Gaurav Kumar
Mohd. Nadeem
Manish Sharma
Neeraj Sharma
Shreya Lanjewar
Ankit Karan
Piyush Goyal

1 OBJECTIVE

To develop a UAV Fleet that can fly autonomously, identify objects from clutter using swarm technology.

2 ABSTRACT

The project requires to develop a UAV fleet that has to find 4 out of 5 mannequins in the field of 40m X 40m. In this task, we are developing a fleet of three quadcopters that relies on master-slave Technology. In this set-up one drone will control the other two drones serving as their communication hub. We will be using openCV for image processing that will be used for object identification.

3 HARDWARE COMPONENTS USED

- Raspberry PI 4B, 4GB RAM
- Flight controller(PixHawk)
- Chassis and Propellers
- BLDC Motor and ESC
- 8MP Stereoscopic camera
- LIPO Battery(3000 mAH)
- Receiver
- GSM module
- nrf24l01 RF module
- GPS module
- Lora wifi module
- NodeMCU ESP8266

4 COMPONENTS DETAILS

- Chassis :- The main frame is glass fiber while the arms are constructed from ultra durable nylon. It weighs around 280g only. Center plates with integrated power distribution PCB to power the ESCs from the battery.
- Specifications - Width : 450mm Height : 55mm Weight : 280g
- Motors :- Brushless (BLDC) outrunner motor is used in our Drone. It is a 1400kV motor. 30A ESCs is used to drive the motor.
  Specifications -
  KV: 1400
  No load Current : 10 V : 0.5 A
  Current Capacity: 12A/60s
  No Load Current @ 10V: 0.5A
  Motor Dimensions: 27.5 x 30mm
  Shaft diameter : 3.175mm.
- Flight controller :- Pixhawk PX4 2.4.7 32Bits Flight Controller for our Quadcopter. It has digital airspeed sensor for fixed-wing aircraft with Telemetry radios with one-mile range for two-way control, telemetry and data logging via free mobile apps or PC/Mac/Linux ground station.
  Specifications -
  32 bit STM32F427 Cortex M4 core with FPU
  168 MHz
  256 KB RAM
  2 MB Flash
  The 32-bit STM32F103 failsafe co-processor

5 COMMUNICATION

NodeMCU ESP8266 would be used on each of the drones.

- The NodeMCU contains WiFi capability and thus can be used to communicate with each other with the help of a central routing, i.e. a laptop with hotspot enabled or an external wifi-router.
- The coordinates of each drone can be updated multiple times by sending a broadcast to 255.255.255.255 including the identification tag of the drone and GPS coordinates. This can also be recorded by the ground system for further

analysis.

6 LOCATION TRACKING

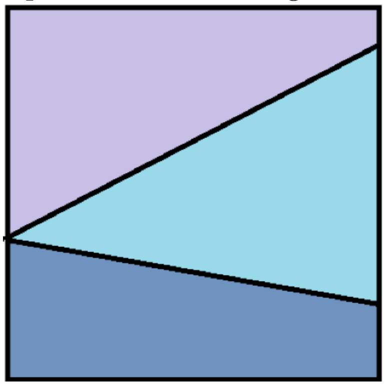
- For this purpose we could use various GPS module compatible with Arduino and RaspberryPi GPIO pins.
- The GPS module generally has a cold start time of a few times and hot start time of a few seconds. Thus, it needs to be calibrated properly pre-flight.
- We would be using Ublox NEO-6M GPS Module as the primary GPS module The module provides an accuracy of 5m, which is not adequate enough in 40m X 40m enclosure, therefore we would have to employ collision avoidance separately to protect the drones.

7 COLLISION AVOIDANCE

- For collision avoidance, we would fit 4 HC-SR04 ultrasonic sensors on all 4 four sides of each drone.
- We would tune the ultrasonic sensor to detect collision in the range which is approximately equal to the precision radius offered by the GPS module.

8 FLIGHT PATH

- As discussed in the forum, the starting point of all the drones would be approximately the same
- Therefore, the aim is to distribute equal area to search for each of the drones without any overlapping of flight paths.
- Also any optimized search algorithm couldn't be implemented because the objects appear randomly in the search space and no fitness of loss function can be deduced. We have to search the whole space.

- Therefore we plan to divide the space in the following manner assuming drones start off from the same point. 
- The dot represents the starting point of drones.
- The partitions divide the square into three parts of equal areas for three drones and can be calculated on the fly using binary search by taking into factor the relative position of the drones to the circle.

9 CAMERA BALANCING

- We are using 2-axis gimbal mechanical tool for balancing the camera in roll and pitch directions.

10 IMAGE PROCESSING

- For object detection, we will use OpenCV on Raspberry Pi with camera located on the drone.
- First, we transform RGB image into HSV plane and the image is converted into binary image after thresholding it on the basis of color.
- After that grouping algorithm is applied on binary image to find groups or connected pixels or in other words green color object.
- The centers of the binary blob in the binary images are computed, and blobs located closer than a predefined min value are merged.
- Finally, center of these objects are calculated. cv2.SimpleBlobDetector() is class provided with OpenCV for step 2 to step 5. A number of are used to reduce Noise before and after the transformation and filtering objects.

11 CURRENT PROGRESS

- Drone is completely assembled.
- We were able to get the drone flying and control it with the transmitter.
- We are able to control the drone using predefined flight paths hard-coded in RaspberryPi.
- The rest of the components would be integrated soon.

12 CONCLUSION

- This project has a vast potential for application in surveillance in difficult terrains which are out of physical reach of people.
- Swarm technology is in its infancy and has a huge potential to be developed further.