Autonomous Drone Path Finder

HS04

Hemanth Vanam - S20170010172
Neelakanta Sriram - S20170010102
Hrithik Puppala - S20170010115

OBJECTIVE

➔ The main aim of our project is to train a group of autonomous UAV or drones to explore a given area and create a virtual map of the area based on the terrain information in the simulation.

First Evaluation:

For the first evaluation ,we have worked on a literature survey . We have read some research papers and we came up with a problem statement and some existing research in that domain.

Then we have done some research on existing simulation tools and decided the tool suitable for simulation (i.e., Unity 3D) .

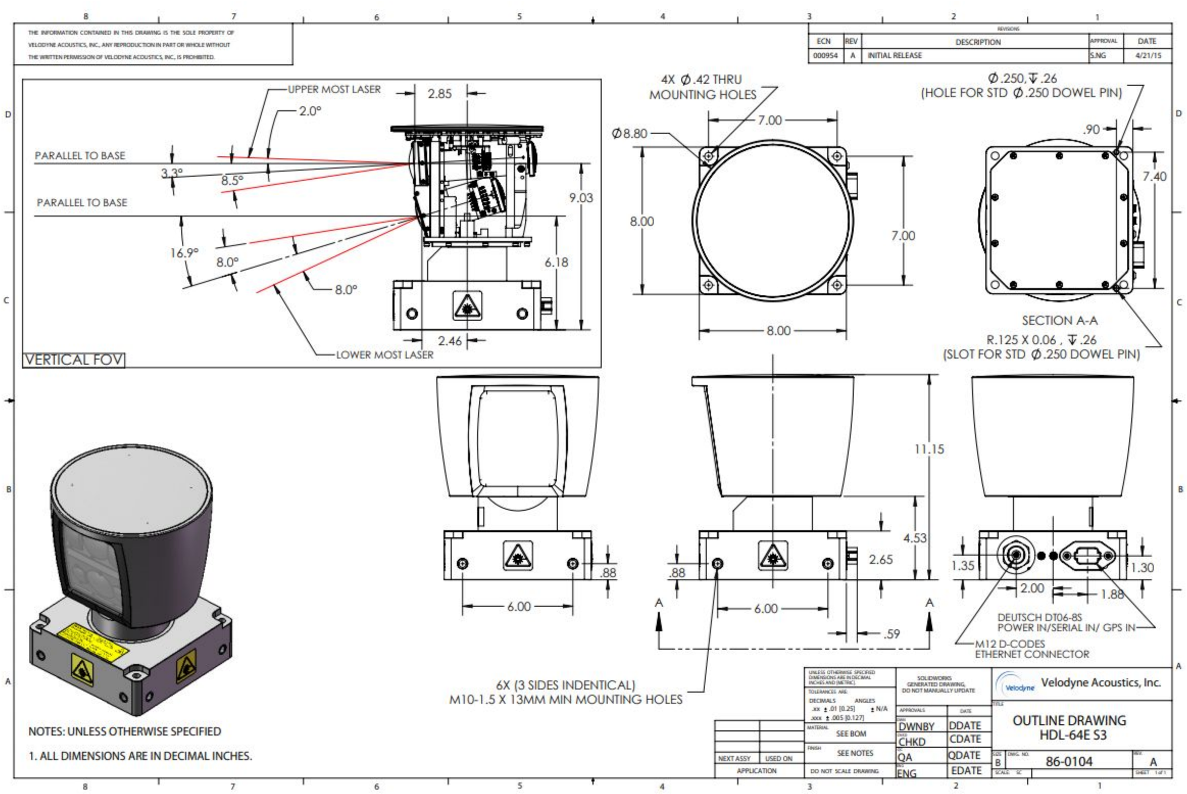
Second Evaluation:

- ➔ We implemented an environment with aim of making a hummingbird model which collects the nectar from the nearest flower and once the nectar of that flower is exhausted move to the next nearest flower.
  - Whenever the episode begins the position of the flower plants are randomly selected.
  - The position of the agent (hummingbird) is randomly selected (a safe position not colliding with any flowers or rocks or other game objects) or simply spawned right in front of a flower bud with a probability of 0.5 for each.
  - A small 0.01 + bonus reward for getting the nectar from the flower.
  - A negative 0.5 reward whenever the agent crosses the boundary of the environment.
- ➔ We have found some drone models on one of which we implemented the training now.

Third Evaluation

Implemented Lidar sensor in Unity:

- Can be done in real time.
- Used the Velodyne HDL-64E model for reference.
- All the cloud point information of the topology information is stored in a file and can be used by other applications for reconstructing the topology information.
- The specifications of the lidar sensor are as described in the image:



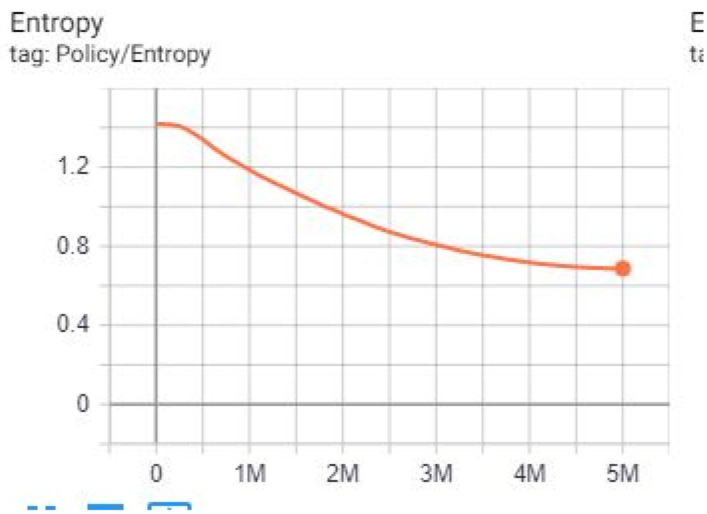
Training Environment for drone movement:

- The environment consists of a custom created terrain of 100x100 square meters with grass and trees.
- Walls of 50 meters height on all sides and roof with collider component.
- Trained the drone using PPO algorithm.
- The input of the NN are:

    - The direction of the target with respect to that of the drone normalised.
    - Drone's local rotation normalised.
    - Dot product of the "toGoal" direction and drone's forward direction.
    - Minimum of distance of the target from the drone and maximum view distance normalised.
    - 6 ray perception sensors in 6 directions of drone.
- The actions of the Drone are:
    - Altitude of the Drone (+1 for Altitude Gain , -1 for Altitude Loss)
    - Yaw of the Drone (+1 for turning left , -1 for turning right)
    - Pitch of the Drone (+1 for tilting forward , -1 to tilting backward)
    - Roll of the Drone (+1 for rolling left , -1 for tilting right)
- Reward System:
    - When the drone is inside the target game object, the reward is calculated based on the exact distance from the goal center to the drone plus a small consistent reward of (+0.01).
    - If the drone collides with any other object, the negative reward (-0.5) is given and the episode stops.
    - If the drone travelled below the terrain or above the roof, a negative reward (-0.5) is given and the episode ends.
- Results:

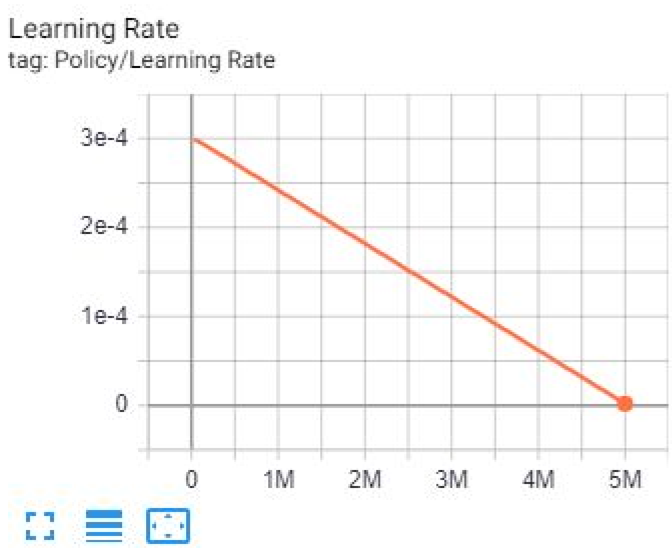
Y -axis : Entropy

X-axis : No. of Steps



Y -axis : Learning Rate

X-axis : No. of Steps



Note: All these results are given in the summaries directories in the project folder

Future Plan

- Create a model which deploys a group of drones and get the terrain information from the drones and generates the virtual map of the area.