Midterm Check-In – UAV Collision Detection & Prevention Project

For our project involving UAV collision detection and prevention, we have made significant progress so far. This is broken up into multiple categories including several improvements to the scene, agent, RL pipeline, lidar sensors, and evaluation metrics. Here, we will describe this progress as well as next-phase goals.

For the scene, we recently added moving objects such as birds and moving cars. These pose a new challenge to our agent because not only does it need to move away from the objects, it also needs to do so in a way that avoids a second move. We are still working on how to get the agent to recognize and properly respond to these moving objects.

For the agent, we are currently working on modeling it to more accurately represent a MK30 Amazon delivery drone. This involves working in a software called Blender to model the drone before implementing the design in Unity. We also are doing research to figure out what values are best for the physics based values of the drone, such as mass, damping, glide, and lift.

For the RL pipeline, each step is performed in the sequence Sensors -> Agent -> Stabilizer -> Actuators. The drone's state information, sourced from its LiDAR sensors and onboard GPS form the input vector which is fed to the agent. The agent will take this vector and using a MLP NN coerces it into a 3 vector output. Each of these vectors represents a different action which the Stabilizer script exposes to the agent: pitch x, roll y, and climb z. The stabilizer executes these instructions from the agent, while also fighting to stay balanced using traditional mathematical algorithms. The stabilizer accomplishes these two goals by outputting N thrust values (in Newtons), one for each rotor the drone has. All the while, a reward function is being tracked by the Agent script which updates the weights of the NN using ordinary backprop at the end of each episode.

For LiDAR, we have upgraded from one sensor to two that emit two hemispheres' worth of beams. There are 144 beams for each sensor, and each is 15 degrees apart. The lasers detect surfaces and record the x, y, z coordinates relative to the drone (where the drone is the origin). Each session is recorded and saved into a separate CSV file within our project in real time. The LiDAR beams also record other data such as velocity, acceleration, pitch, yaw, and roll. For each beam, it records the euclidean distance for each beam, the azimuth angle, the elevation angle and if it scans an object/surface. For the future, we will implement motor strength for each propeller, currently there are only 4 but it is increasing to 6 with our new model for the drone which better represents the Mk30 drone. We also plan on using the data recorded in our csv files to help train the drone for better collision detection.

The evaluation metrics we are currently using include a mean reward function, which is calculated based on collisions, step count, success rate, proximity/distance to goal, and whether the flight is realistic. (does not have too much tilt, takes too long, etc) We currently are averaging around 11 for this mean reward, which is not great considering a success adds 50.

Our goals for the next-phase include improving the realism of the environment, the quality of the training pipeline, and the actuators. We also want to add evaluation metrics such as success rate over multiple episodes, crash rate, and efficiency. If time allows, we want to adapt our agent to handle interaction with other agents as well.