EN.525.637.81.FA23 —- Foundations of Reinforcement Learning Project Proposal: A curriculum learning approach to coordinated multi-agent drone delivery systems

Alec Portelli

1. Background & Motivation

In a world where there is an increasing demand to get shipments around as quickly as possible, drones eliminate many of the problems faced by modern delivery systems. There are a lot of logistical challenges when it comes to moving goods from place to place, and it is often difficult to find an optimal solution that is time and cost effective. Drones offer a unique solution because they are fast, agile, reliable, and are environmentally friendly [2].

However, despite all of these benefits, building autonomous delivery systems is incredibly challenging. Many drone systems have manual path planning, GPS, and other tools that help them navigate a deterministic environment. But more often than not these tools stop functioning in a real world scenario and leave the drone useless without any sort of autonomy [1]. Introducing reinforcement learning (RL) to these drones to build a full autonomy stack allows users to deploy a drone and not have to worry about depending on outside tools to help it navigate the world [1]. Furthermore, RL can assist in path planning, mapping, and coordination with other drones to ensure optimal performance. An autonomous delivery system would have a large amount of applications, ranging from consumer goods all the way to military transport. It would also cut down on carbon emissions from delivery trucks and save costs on delivery drivers.

But even with modern day RL techniques, it is a big assumption to put drone into a real world scenario and hope that it will being to learn quickly. The environments that a drone would encounter on an average delivery are too difficult to start training on, which is why curriculum learning is needed. Curriculum learning gives the agent a chance to work up to the most demanding environments by slowing increasing difficulty. A robust RL model needs to be trained on the most realistic environments as possible, which is why curriculum learning is perfect for this RL challenge.

2. Problem Modeling

This project aims to simulate realistic scenarios for drones to drop off a package and return to their home location. The drone agent will be trained using RL algorithms using a curriculum learning approach.

Due to the large amount of factors that come with replicating a real world environment, this project is focusing on navigation and multi-agent training. To simplify the problem, a very simple rigid body kinematics model will be used to simulate the drone's motion. People will also be removed from the training environments, although there will be constraints on where the drone can and cannot go to mimic not hitting humans on the street.

The drone will be equipped with a simple LiDAR for spatial awareness and a IMU for keeping track of orientation and velocity. While control is not the focus of this project, the drone still needs position and orientation to navigate the environment.

The RL problem can be broken down into the following:

1) Agent: The drones carrying the delivery

2) Environment: The 3D world the drone has to navigate. The environments will become increasingly difficult as the drone performs better. Domain randomization will be utilized so the model does not overfit onto one environment.

3) Action Space: A discrete action space will be utilized to control the drone. The drone can translate in all three world axis and also needs an action to drop off the package.

4) State Space: The observations that describe the drone in a point in time during the simulation. The state space includes: the drone, home, and end goal position in XYZ, the drone velocity, and the four ray casts of the LiDAR for positioning.

5) State Transition: How the agent goes from the action taken from the policy in the environment to now having an updated state after the action has been taken. This will occur every time the drone makes a decision and moves to a new position.

6) Reward: The feedback given to the agent based on if the taken action was a good decision or not. The major rewards are given to the agent if it gets to the goal, drops off the delivery, and returns back to base. The drone gains points as it stays in control and takes as few moves as possible to accomplish the task. The drone loses a lot of points by crashing into obstacles and other drones, or not being able to find the goal.

3. Methods

Unity Engine is the chosen simulator to train the drones in. Unity provides a physics engine that includes rigid body kinematics, LiDAR, and colliders. Unity also has the ML-Agents plugin, which has all the components needed to do RL with custom agents.

The algorithm of choice for this project is PPO, which is the standard for the Unity ML-Agents package. PPO has shown to have great success in multi-agent reinforcement learning, which makes it a great choice for this project [3].

To begin training the drones, Unity will generate some simple 3D environments. The goal of the agent is to deliver the package regardless of the complexity of the environment. Domain randomization will be utilized to create a new training environment every episode to create a more robust model and to prevent overfitting.

As the drones begin to perform beyond a certain threshold, the environments will begin to become more difficult. A curriculum learning YAML file will be configured to determine the thresholds and specify how much harder the environments get over time.

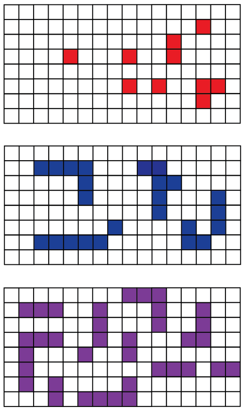

Figure 1. Example on how training environments could get harder over time

The average reward of the episode is kept track of over time, which is what will trigger whether or not the difficulty needs to be increased. However, if the agent is stuck on a certain difficulty, the complexity of the environment can be simplified down a level to give the agent a chance to get some missing experience. Checkpoints will be assigned during the training process to ensure that the correct level of difficulty is being generated, along with giving datapoints needed for algorithmic evaluation.

Along with domain randomization, Unity provides the infrastructure necessary for parallel training environments. This is important because the agent can see many different environment configurations at once, allowing it to build a more robust model. This will also give clearer insight as to what the reward totals for that episode mean, as the agent is not training on the exact same environment every episode.

4. Evaluation

With the curriculum learning schema, domain randomization, and parallel training environments, the agent is going to see a wide variety of environments. To accurately evaluate the agent, five handmade difficult environments will be made. The agents will spawn in as per usual and are expected to complete the task without any failures.

A handcrafted environment ensures it will be slightly different than the ones generated, which will test how robust the model is. The agent will be evaluated on the score it gets versus the max score, needing to pass the threshold of 80%. Unity also provides a connection to TensorBoard while training, so the statistics on how the model has done over time can also be evaluated.

5. Experiment Aims

The experiment aims to build a simulation environment using Unity to train a drone using PPO. The environments will be generated by Unity and the difficulty will be managed by a curriculum learning configuration. All of these components create a testbed that can be used to train and evaluate drone delivery models.

After training has been completed, a model evaluation will be conducted. The aim is to train a model to navigate a custom built environment to show that RL can handle all of the complexities that a real world delivery scenario will provide.

References

[1] Muñoz, Guillem, Cristina Barrado, Ender Çetin, and Esther Salami. 2019. "Deep Reinforcement Learning for Drone Delivery" *Drones* 3, no. 3: 72. https://doi.org/10.3390/drones3030072

[2] Bi, Zhiliang, Xiwang Guo, Jiacun Wang, Shujin Qin, and Guanjun Liu. 2023. "Deep Reinforcement Learning for Truck-Drone Delivery Problem" *Drones* 7, no. 7: 445. https://doi.org/10.3390/drones7070445

[3] Yu, Chao, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." arXiv, eprint 2103.01955. Published in 2022.