Machine Learning Engineer Nanodegree

Capstone Proposal

Said Aboulaban

August 23rd, 2018

# Proposal

## Domain Background

The project proposed has a main focus on Reinforcement Learning, focusing on Autonomous Quadcopter tasks. Autonomous UAVs has been a field of study for quite a long time, and there were major breakthroughs in the field, ranging from Model Free Actor/Critic methods; e.g. <https://arxiv.org/abs/1509.02971>, trust Region Policy Optimization; e.g. <https://arxiv.org/pdf/1502.05477.pdf>, or a combination of both; e.g. <https://arxiv.org/pdf/1707.05110.pdf>. One of the interesting areas of studies is UAV's self-protection through motion maneuvers. The idea is inspired by bird's ability to avoid threats, through motion maneuvers by taking decisions in split seconds. The project attempts to train the UAV to stabilize in air; in real life that can be for a specific task like imaging, and through motion maneuvers react to actions targeted at it by untrusted moving bodies. This work will be part of a personal project to work on an UAVs for children safety, that can go into a "panic mode" when feels threatened.

## Problem Statement

Drones are becoming more popular for personal and business use, and still are pricey appliances. With wide range of applications like imaging for personal use, football matches imaging, home security, drones are even used in Telecom industry to perform tasks risky for humans like communication tower inspection. When operating in real world, there can be a lot of moving or static hazardous objects that can damage the drone while the UAV is moving or in steady state.

## Datasets and Inputs

Since this is a Reinforcement Learning project no datasets will be used, however, flight and environment simulators will be used. The initial Quadcopter simulator used will be the physics\_sim.py located at: <https://github.com/udacity/RL-Quadcopter-2>. More advanced simulators can be used in the future if time permits. Environment class will be developed as part of this project.

Environment class will be based on simple indoor setup, and will be constructed randomly at the beginning of every episode. Objects considered for the environment, indoor basic setup, static objects randomly placed, target object; with a fixed fly zone around it, where the drone needs to stay in, can be the target object of imaging, or inspection or any other use, Untrusted Moving Objects that are hazardous to the UAV.

## Solution Statement

The UAV will always start the episode from a random 3D point in the fly zone. The UAV will remain steady in its location, unless “threatened” by a hazardous object moving with a risk of collision. Once the drone is threatened, it should hop using thrust to a new 3D location to avoid the hazardous object, and remain steady in the new location.

## Benchmark Model

Since the agent is planned to learn multiple tasks, different benchmark models will be used. For navigation to a target within a flying zone, the following study will be used: <https://arxiv.org/abs/1509.02971>, and the main benchmark will be time to convergence. For collision detection and avoidance, the following study will be used: <https://arxiv.org/pdf/1702.01182.pdf>, and the benchmark will be the number of collisions per speed.

## Evaluation Metrics

As highlighted in the previous section, the following benchmarks will be used:

1. Average sum of rewards (costs) and its convergence
2. Number of crashes due to collision at different speeds.

Collision benchmarking will be based on different scenarios:

1. Collision with static objects during hopping
2. Collision with dynamic objects while drone airborne in steady state
3. Collision with static and dynamic objects while drone is airborne; hopping or steady state

## Project Design

### Phase I: Environment Class creation and testing

During this phase, the focus will be on creating an env class to address the different env scenarios

### Phase II: Actor/Critic/Agent Design:

During this phase, the plan is to implement the following classes:

1. Risk Averse Network: based on https://arxiv.org/pdf/1702.01182.pdf

- Input Layers: Current Position, Distance to collision objects, Relative velocity to collision objects, Actions performed

- Hidden Layers:

- Hidden Layer 1: 40 units, Activation RELU

- Hidden Layer 2: 40 units, Activation RELU

- Output Layer:

- Collision Probability, None x 1, Activation Sigmoid

- Bootstrap: 5

- Dropout Ratio: 0.05

- Average of the 5 models is used to estimate along with a hyper-parameters lambda-std, and sample from that distribution a collision probability.

1. Critic Class: Will be based on function estimation for value function, implementation will be based on tensorflow

Expected Architecture is: based on https://arxiv.org/abs/1509.02971. The modification proposed is to add collision probability as an input into the critic networks

- Input Layers: Current Position, signals, current linear/angular velocities, Objects in proximity, their velocity, shape and orientation, Last Action

- Hidden Layers:

- Action Hidden Layer 1: 32 units, Activation RELU

- Action Hidden Layer 2: 64 units, Activation RELU

- State Hidden Layer 1: 32 units, Activation RELU

- State Hidden Layer 2: 64 units, Activation RELU

- Output Layer:

- Action Value Function Estimate: None x 1, Activation None

1. Actor Class: based on https://arxiv.org/abs/1509.02971. The modification proposed is to add collision probability as an input into the critic networks

- Input Layers: Current Position, signals, current linear/angular velocities, Objects in proximity, their velocity, shape and orientation

- Hidden Layers:

- Hidden Layer 1: 32 units, Activation RELU

- Hidden Layer 2: 64 units, Activation RELU

- Hidden Layer 3: 32 units, Activation RELU

- Output Layer:

- Rotor Thrust

1. Agent:

- The agent will be based on <https://arxiv.org/abs/1509.02971> with modification of including collision probability. And if time permits, will test advantage function as a critic output into the actor class.

1. Task: Main tasks will be created

- Stabilize and avoid moving obstacles

- Reward function is planned as per the following:

- Episode Ends with high penalty: if Collision with any object

- Episode Ends with high reward: if flight time ends with no collision and the drone in the flight zone

- Incremental reward wrt time stayed in place while airborne in fly zone

- Incremental penalty for every time step stayed in non-fly zone

- Incremental penalty wrt collision probability

### Phase II: Benchmarking:

1. Scenario 1: No objects in the environment, objective is to test the UAV’s ability to stay in steady state; and optimize the reward function related to steady state in fly zone. KPI will be rewards time to converge
2. Scenario 2: Single Static Object, objective is to test the UAV’s ability to stay in steady state, while observing a static object, and optimize the reward function related to collision probability. KPI will be rewards time to converge
3. Scenario 3: Single Moving Object, objective is to test the UAV’s ability to avoid moving obstacles; and optimize the reward function related to collision probability. KPI will be number of collisions at different relative speed
4. Scenario 4: Single Moving Object and Single Static Object, objective is to test the UAV’s ability to maneuver knowing that there are obstacles in the environment that it needs to avoid during maneuver. Main KPIs will be rewards convergence time, and collisions per speed.
5. Scenario 5: This will be assessed if time of the project permits, with multiple moving objects and multiple static objects. Objective is to test real world scenario. Main KPIs will be rewards convergence time and collisions per speed.