Technische Universität München
Fakultät für Elektrotechnik und Informationshtechnik
Lehrsthul für Datenverarbeitung

Project Proposal – Group F

Game of Drones

Domenico Tomaselli, Mat.–Nr. 03677426
Michael Seegerer, Mat. –Nr. 03676417
Sayanta Roychowdhury, Mat. –Nr. 03709791

23.04.2020 – 07.05.2020

1 Motivation

While the use of unmanned aerial vehicles (UAVs) originated mostly for military purposes, their application towards other fields is rapidly growing and improvements in the technology of the aircraft represent an active area of research. On the current market, drones equipped with high definition cameras can be easily purchased for a broad range of applications, nevertheless their control mechanism remains a challenging aspect of the technology if traditional engineering techniques are employed. The application of ML approaches such as Reinforcement Learning for such task could potentially result in the successful design of an algorithm/agent capable of learning how to operate the drone without human supervision and spare the effort currently invested in developing and implementing traditional controlling schemes.

There is a growing consensus in the aircraft industry to reduce the pollution caused by the aircraft; according to the European Aviation Environmental Report for 2019, between 1990 and 2016 the CO_2 emissions of all flights departing from EU28 and EFTA has increased from 88 to 171 million tonnes (+95%). The application of Reinforcement Learning could have the potential to facilitate the alignment of UAVs to such consensus by expanding the design of the algorithm/agent to optimize the fuel consumption while being airborne and navigating to target locations [1].

Last but not least, the take–off mechanism of an UAV, which will be one of the central tasks required to be performed by our agent, remains an appealing topic from several perspectives; it is not merely a theory–based problem, but it has substantial use–cases in practise and it is at first glance an operation characterized by a fairly restricted set of actions or rather a limited set of factors influence its mechanism.

2 Goals

The main objective of our project is to design an environment capable of simulating the take–off mechanism of an UAV and utilize it to investigate and construct control procedures for such operation according to the different aspects and factors influencing it.

The design of our environment must equate the quintessential physical and financial aspects that characterize a standard UAV controlling procedure, including the thrust generated from the different engines and its counteracting gravitational force as well as potential disturbing factors such as the presence of directional wind forces or poor reception of the UAV's wireless controller. From a financial standpoint, our goal is to incorporate action patterns that influence the costs of operation such as the fuel/battery consumption resulting from operating the UAV and the time efficiency.

From a more technical point of view, the goal of our project is to construct a two–dimensional environment simulating an airborne UAV in earth–like surroundings. The agent must be capable of performing the take–off mechanism in safe fashion using separately controllable engines located at its two bottom corners and under the influence of the counteracting gravity pull; once the take–off is complete, the agent must navigate the drone to a target location specified a priori using the least amount of fuel/battery capacity possible and once such location is reached it must be maintained in stable fashion for a number of time steps.

Regarding the prototype and the initial task to be performed, the goal is to limit the complexity of the environment to be solved by only attempting the take–off of the UAV from its starting ground position and reach a target location placed vertically above it. Once the initial task is successfully completed, the complexity will be increased by shifting the target location left or right respect to the take–off position of the drone. Investigating secondary control aspects of the UAV such as optimizing the consumption of fuel/battery capacity or increasing the time efficiency will not be included in the prototype, instead it will be implemented during the main working phase.

3 Steps

Our initial approach to the project will consist of familiarizing ourselves with the tools required for the implementation of the designed environment, i.e. Box2D and the existing templates available in OpenAI. The core idea behind the take–off procedures and subsequent control of an UAV derives from one of the OpenAI environments, i.e. the *LunarLander* [2] and to start off we will dedicate some time to examine its implemented features and procedures. Once we acquire a certain level of confidence with the structure of the template, we will proceed to learn how to operate the simulation tool Box2D and design our prototype with it.

3.1 State space

To constraint the complexity of the environment the state space was restricted to a discrete domain.

State space prototype:

- Location (x_t, y_t): Current position of the drone's left bottom corner in xy–coordinates within 20x20 grid.
- Angle ϕ_t: Current angular relation of the bottom platoon to the ground $\rightarrow$ discretized in steps of 15 degrees

Advanced options for future developments:

- Velocity (v_{x_t}, v_{y_t})

Regarding the environment's prototype, the state space will contain the xy–coordinates of the drone's left bottom corner and the angle computed in relation to the ground; according to these information, we estimate that a complete description of the drone's location within the 20x20 grid can be accessed. From a technical point of view, such description will translate in terms of strict algorithmic implementation into a three–dimensional array whose dimensions are defined as follows: (i) the first dimension describes the x–coordinate of the left bottom corner, (ii) the second dimension describes the y–coordinate of the left bottom corner and (iii) the third dimension describes the estimate of the angle.

During the project's main working phase, the complexity of the state space will be incremented by introducing the velocity factor to reinforce the correctness of the environment's physical behaviour.

3.2 Action space

Regarding the action space, our initial approach for the prototype is to restrict it to two possible actions for each of the drone's engine:

Action space prototype:

- $u_{t,i} = 1$ if left $(i = 1)$/right $(i = 2)$ engine activated and
- $u_{t,i} = 0$ if left $(i = 1)$/right $(i = 2)$ engine shut down

Advanced options for future developments:

- $u_{t,i} \in N$ and $i \in \{1, 2\}$

Considering the downsampling of the environment from three to two spatial dimensions, the number of engines employed for the drone's navigation is reduced from the standard four engines to a two engines configuration. Based on this design, the agent/algorithm will be prompted to undertake one of four different combinations of actions at each time step t in the attempt to take–off the UAV and navigate it to the target location: (i) activate both right and left engines to enforce a greater thrust than the counteracting gravitational force and induce the drone's ascending, (ii) activate either the right or the left engine to enforce a greater thrust only on the side of the drone where the engine is initiated or (iii) shut down both right and left engine generating not enough thrust to withstand the counteracting gravitational force and causing the drone's descending.

The expansion of the action space from a strictly Boolean domain to a discrete domain of possible degrees of thrust for each engine will be further investigated and implemented during the main working phase.

3.3 State transitions

Regarding the transition from the current state to the next state, our initial approach is as follows:

State transitions:

- Left and right engines are shut down $u_{t,i} = 0, \forall i$: Gravitational pull forces the drone to descend by one cell.
- Left and right engines are activated $u_{t,i} = 1, \forall i$: Engines running at full capacity, generated thrust leads to drone's ascending by one cell.
- Left engine activated $u_{t,1} = 1$/right engine shut down $u_{t,2} = 0$: left engine running at full capacity propels drone's left bottom corner up by one, while the gravitational pull forces drone's right bottom corner to descend by one since no thrust. The established angle ϕ_{t+1} between the two bottom corners is estimated according to

$$\phi_{t+1} = \arcsin\left(\frac{y_{right} - \Delta h}{w}\right), \tag{1}$$

  where y_{right} is the y–coordinate if the drone's right bottom corner, Δh is the height difference that creates itself between the two bottom corners and w is the drone's width.
- Left engine shut down $u_{t,1} = 0$/right engine activated $u_{t,2} = 1$: right engine running at full capacity propels drone's right bottom corner up by one, while gravitational pull forces drone's left bottom corner to descend by one since no thrust. The established angle ϕ_{t+1} between the two bottom corners is estimated according to

$$\phi_{t+1} = \arcsin\left(\frac{y_{right} + \Delta h}{w}\right). \tag{2}$$

Advanced options for future developments:

- Wireless communication $\mathrm{P}(U)$: process of taking actions expanded as discrete probability distribution

$$\mathrm{P}(U = u_{t,i}) = 0.8, \quad \mathrm{P}(U = random(u_{t,i})) = 0.2 \tag{3}$$

$$0 < \mathrm{P}(U) < 1, \quad \sum \mathrm{P}(U) = 1. \tag{4}$$

- Turmoil caused by climatic agents: wind can affect by the drone's current route by increasing its current velocity/producing undesired shifts of its current position across the grid.

To incorporate potential disturbances of the wireless communication between the drone and the remote control, our initial idea is to formulate the process of taking an action as a discrete probability distribution $\mathrm{P}(U)$ with U as the random variable denoting which action will be ultimately carried out by agent, either the designed action (with a probability of 0.8) or a random action (with a probability of 0.2).

3.4 Reward function

An initial approximation of what the environment's reward function could resemble is as follows:

Reward function:

$$r(s_t, u_{t,i}) = r(x_t, y_t, \phi_t, u_{t,i}) = \begin{cases} -1, & \forall t \in T \\ -50, & \text{if } (x_t, y_t) = (x_{ground}, y_{ground}) \\ +100, & \text{if } (x_t, y_t) = (x_{target}, y_{target}) \\ -5, & \text{if } u_{t,i} = 1 \text{ and } i \in \{1, 2\} \end{cases} \tag{5}$$

where T is the total number of time steps, (x_{ground}, y_{ground}) is the drone's starting position on the ground and (x_{target}, y_{target}) is the target location to be reached.

The agent will receive an overall punishment (negative reward of -1) for every time step of the episode, while compensation (positive reward of +100) will only be awarded once the target area is reached; an additional sanction (negative reward of -50) will be enforced for every time step the agent is not capable of lifting the drone from the ground, as one of its main objectives remains the successful completion of the drone's take–off mechanism.

To enforce the optimization of fuel/battery consumption by the agent, the reward function was extended to incorporate an additional punishment (negative reward of -10) at every time step one of the two engines is activated. The idea is to train the agent to reach the desired target location by activating the engine only if unavoidable to preserve as much fuel/battery capacity as possible while completing the task within the environment.

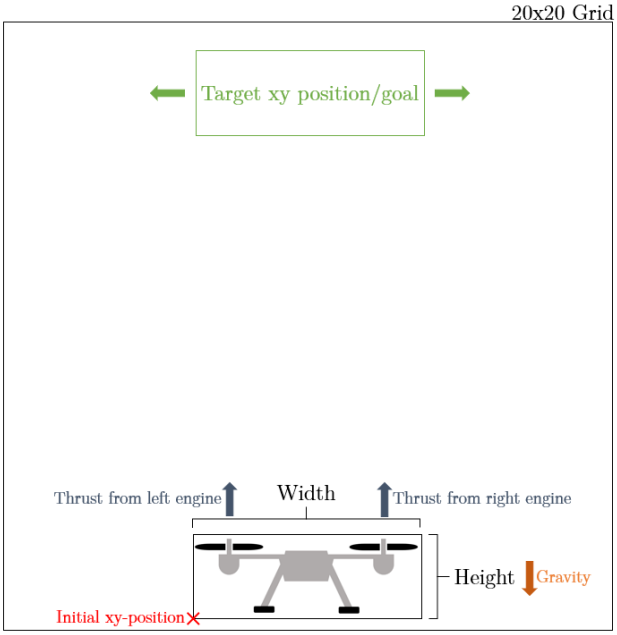

Figure 1: Initial sketch of the environment; the drone starting from a ground position must take-off and reach a priori defined target position within a 20x20 grid.

4 Time Plan

The time plan for the project is visualized in Figure 2 in terms of the weeks of work projected to complete each task; in addition to that, we have set up a more detailed schedule in Figure 3 for the first two milestones distributing the workload among the members of our group.

		Weeks of work projected	
		One week	Two weeks
Prototyping & Testing	Analysis		
	Prototype's design & implementation		
	Testing (implement test agent)		
Main Working Phase	Improve environment design		
	Implement RL agent		
	Solve environment		
	Testing & debugging		
Peer Review Phase	Apply RL to three environments		
	Write results & final report		
Final Presentation	Prepare final presentation		

Figure 2: Time plan containing the weeks of work projected to complete the various tasks for the project's implementation.

			Phase 1: Submission project proposal														Phase 2: Prototyping of environment design																									
			Week 1							Week 2							Week 3							Week 4							Week 5							Week 6						
Task Description	Task Owner	Due Date	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
Motivation of the project	Sayanta	04/05/2020																																									
Goals of the project	Michael	04/05/2020																																									
Describing the project	All	04/05/2020																																									
Timeline of the project	Domenico	04/05/2020																																									
Initial simulation Box2D	All	10/05/2020																																									
Initial rendering according to OpenAI Gym	All	10/05/2020																																									
Implementing action function	Domenico	31/05/2020																																									
Implementing reward function	Michael	31/05/2020																																									
Implementing next state function	Sayanta	31/05/2020																																									
Rendering for model visualization	Domenico	31/05/2020																																									
Design a sample agent and test environment	Michael & Sayanta	31/05/2020																																									
Writing intermediate report	All	04/06/2020																																									
Presentation of currrent project status	All	04/06/2020																																									

Figure 3: Time plan distributing the workload among our group for the first two milestones.

References

[1] E. E. A. (EEA), E. U. A. S. A. (EASA), and EUROCONTROL, "European aviation environmental report," 2019.

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.