A Simple Platform for Reinforcement Learning of Simulated Flight Behaviors *

Simon D. Levy

Computer Science Department,
Washington and Lee University,
Lexington VA 24450, USA
simon.d.levy@gmail.com

Abstract. We present work-in-progress on a novel, open-source software platform supporting Deep Reinforcement Learning (DRL) of flight behaviors for Miniature Aerial Vehicles (MAVs). By using a physically realistic model of flight dynamics and a simple simulator for high-frequency visual events, our platform avoids some of the shortcomings associated with traditional MAV simulators. Implemented as an OpenAI Gym environment, our simulator makes it easy to investigate the use of DRL for acquiring common behaviors like hovering and predation. We present preliminary experimental results on two such tasks, and discuss our current research directions. Our code, available as a public github repository, enables replication of our results on ordinary computer hardware.

Keywords: Deep reinforcement learning · Flight simulation · Dynamic vision sensing.

1 Motivation

Miniature Aerial Vehicles (MAVs, a.k.a. drones) are increasingly popular as a model for the behavior of insects and other flying animals [3]. The cost and risks associated with building and flying MAVs can however make such models inaccessible to many researchers. Even when such platforms are available, the number of experiments that must be run to collect sufficient data for paradigms like Deep Reinforcement Learning (DRL) makes simulation an attractive option.

Popular MAV simulators like Microsoft AirSim [10], as well as our own simulator [7], are built on top of video game engines like Unity or UnrealEngine4. Although they can provide a convincingly realistic flying experience and have been successfully applied to research in computer vision and reinforcement learning, the volume and complexity of the code behind such systems can make it challenging to modify and extend them for biologically realistic simulation.

For use in biologically realistic flight modeling, however, simulators like these, suffer from two important limitations. First, video game engines are designed to

* The author gratefully acknowledges support from the Lenfest Summer Grant program at Washington and Lee University, and the helpful comments of two anonymous reviewers.

run in real time for human interaction, making them orders of magnitude too slow to collect sufficient samples for most reinforcement learning algorithms. Second, the use of photo-realistic data rendered at a standard video game frame rate (60 - 120 fps) makes them suitable for modeling data acquisition by actual video cameras, but unsuitable for the low-resolution, fast/asynchronous visual sensing typical of insects and other organisms [8].

2 OpenAI Gym Environment

OpenAI Gym [2] is a toolkit for developing and comparing reinforcement learning algorithms. In addition to providing a variety of reinforcement learning environments (tasks) like Atari games, it defines a simple Application Programming Interface (API) for developing new environments. Our simulator implements this API as follows:

- `reset()` Initializes the vehicle's state vector (position and velocity) to zero.
- `step()` Updates the state using the dynamics equations in [1].
- `render()` Shows the vehicle state using a Heads-Up-Display (HUD) or third-person view.

In the remainder of this extended abstract, we discuss related projects based on OpenAI Gym and provide a brief overview of work-in-progress on a new simulator designed to address these issues in that framework.

3 Related work

Ours is one of a very small number of published, open-source projects using OpenAI Gym to learn flight-control mechanisms. Two others of note are (1) Gym-FC, which uses OpenAI Gym and DRL to tune attitude controllers for actual vehicles [6], and (2) Gym-Quad, which has been successfully used to learn landing behaviors with spiking neural nets using a single degree of freedom of control [4]. Because we wished to explore behaviors beyond attitude-control and landing, we found it more straightforward to build our own Gym environment, rather than attempting to modify these already substantial projects.

4 Results to Date

Speedup

Without calling OpenAI Gym's optional `render()` function, we are able to achieve update rates of around 28 kHz on an ordinary desktop computer – an order of magnitude faster than the 1 kHz rate reported for AirSim [10], and approximately three times fast as the rate we observed with our own UnrealEngine-based simulator.

Learning

As a simple test of applying our platform to a reinforcement-learning task, we used a Proximal Policy Optimization (PPO) agent [9] to learn a challenging behavior, using a dynamics model approximating the popular DJI Phantom quadcopter.

In this behavior, *Lander2D*, the agent was given the vehicle's horizontal and vertical coordinates, their first derivatives, and the vehicle's roll and its first derivative (six degrees of freedom). The control signal consisted of roll and thrust commands (two degrees of freedom). As with the popular Lunar Lander game, the goal was to land the vehicle between two flags in the center of the landscape, with a large penalty (-100 points) for flying outside the landscape, a large bonus (+100 points) for landing between the two flags, and a small penalty for distance from the mid-point, to provide a gradient. (See Figure 1 (a) for an illustration.)

Our preliminary results were encouraging. In the *Lander2D* behavior, the PPO agent learned to land the vehicle in under 9,000 training episodes (around 17 minutes on a Dell Optiplex 7040 eight-core desktop computer with 3.4GHz processors). The best score achieved by the agent was competitive with the score obtainable through a heuristic (PID control) approach, around 205 points.

Our current work involves three directions: (1) replacing the PPO agent with a more biologically realistic spiking neural network (SNN); (2) extending our results to a full three-dimensional simulation, as shown in Figure 1 (b); and (3) adding a simulated vision system.

For the third direction – adding visual observation – DRL algorithms have traditionally used convolutional neural network (CNN) layers to enable the network to learn the critical features of the environment from image pixels. CNNs have however come under criticism for lacking biological plausibility and requiring very large amounts of computation time to learn the relationship between the pixels and the world state [5].

To address these issues in a biologically plausible way, we are developing a simple simulation of an event-based vision sensing [8] to incorporate into our platform. Instead of generating simulated events by sampling of an ordinary camera image, our event simulator uses a rudimentary perspective model to generate pseudo-events based on the vehicle state and the position of a simulated target object, represented as a sphere. We plan to use this simulator to model visually-guided predation with SNNs, extending the SNN landing-behavior work presented in [4].

5 Code

Our code, with instructions for reproducing our results, is available at `https://github.com/simondlevy/gym-copter`.

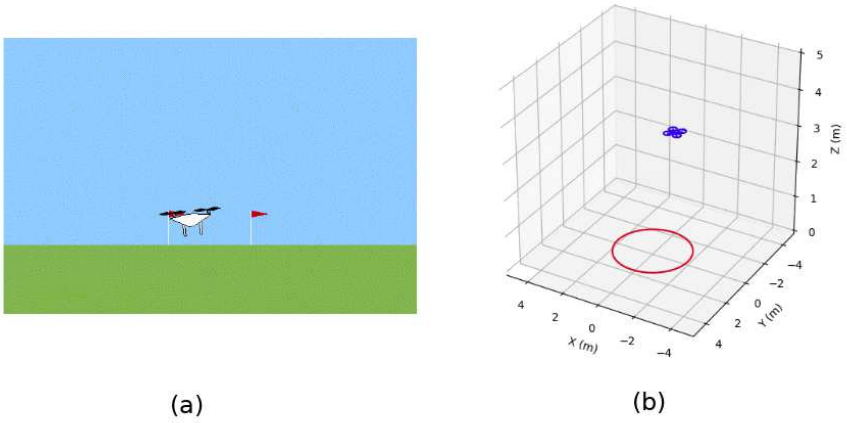

Fig. 1. 2D (a) and 3D (b) environments for the Lander task.

References

1. Bouabdallah, S., Murrieri, P., Siegwart, R.: Design and control of an indoor micro quadrotor. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004. vol. 5, pp. 4393–4398 Vol.5 (2004)
2. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. CoRR **abs/1606.01540** (2016), http://arxiv.org/abs/1606.01540
3. Cope, A.J., Ahmed, A., Isa, F., Marshall, J.A.R.: Minibee: A minature mav for the biomimetic embodiment of insect brain models. In: Martinez-Hernandez, U., Vouloutsi, V., Mura, A., Mangan, M., Asada, M., Prescott, T.J., Verschure, P.F. (eds.) Biomimetic and Biohybrid Systems. pp. 76–87. Springer International Publishing, Cham (2019)
4. Hagenaars, J.J., Paredes-Valls, F., Boht, S.M., de Croon, G.C.H.E.: Evolved neuromorphic control for high speed divergence-based landings of mavs (2020)
5. Hinton, G.: What is wrong with convolutional neural nets ? (Dec 4 2014), MIT Brain and Cognitive Sciences Fall Colloquium Series
6. Koch, W., Mancuso, R., West, R., Bestavros, A.: Reinforcement learning for uav attitude control. ACM Transactions on Cyber-Physical Systems **3**(2), 22 (2019)
7. Levy, S.D.: Robustness through simplicity: A minimalist gateway to neurorobotic flight. Frontiers in Neurorobotics **14**, 16 (2020). https://doi.org/10.3389/fnbot.2020.00016
8. Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., Delbruck, T.: Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. Proceedings of the IEEE **102**(10), 1470–1484 (2014)
9. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017), https://arxiv.org/abs/1705.05065
10. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2017), https://arxiv.org/abs/1705.05065