

Autonomous Flight Arcade Challenge: Single- and Multi-Agent Learning Environments for Aerial Vehicles

Extended Abstract

Paul Tylkin
Massachusetts Institute of Technology
ptylkin@csail.mit.edu

Kyle Palko
U.S. Air Force AI Accelerator
kyle.palko.1@us.af.mil

Tsun-Hsuan Wang
Massachusetts Institute of Technology
tsunw@csail.mit.edu

Ross Allen
MIT Lincoln Laboratory
ross.allen@ll.mit.edu

Tim Seyde
Massachusetts Institute of Technology
tseyde@csail.mit.edu

Alexander Amini
Massachusetts Institute of Technology
amini@csail.mit.edu

Daniela Rus
Massachusetts Institute of Technology
rus@csail.mit.edu

ABSTRACT

The Autonomous Flight Arcade (AFA) is a novel suite of single- and multi-agent learning environments for control of aerial vehicles. These environments incorporate realistic physics using the Unity game engine with diverse objectives and levels of decision-making sophistication. In addition to the environments themselves, we introduce an interface for interacting with them, including the ability to vary key parameters, thereby both changing the difficulty and the core challenges. We also introduce a pipeline for collecting human gameplay within the environments. We demonstrate the performance of artificial agents in these environments trained using deep reinforcement learning, and also motivate these environments as a benchmark for designing non-learned classical control policies and agents trained using imitation learning from human demonstrations. Finally, we motivate the use of AFA environments as a testbed for training artificial agents capable of cooperative human-AI decision making, including parallel autonomy.

KEYWORDS

Deep Reinforcement Learning, Aerial Vehicles, Learning Environments

ACM Reference Format:

Paul Tylkin, Tsun-Hsuan Wang, Tim Seyde, Kyle Palko, Ross Allen, Alexander Amini, and Daniela Rus. 2022. Autonomous Flight Arcade Challenge: Single- and Multi-Agent Learning Environments for Aerial Vehicles: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Developing artificial agents capable of successfully solving problems that are of interest to humans requires designing environments for the agents to learn in. Training AIs directly in the real world can be prohibitively expensive and unsafe, and does not allow for systematic experiments with rare events or unexpected surprises.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online. © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

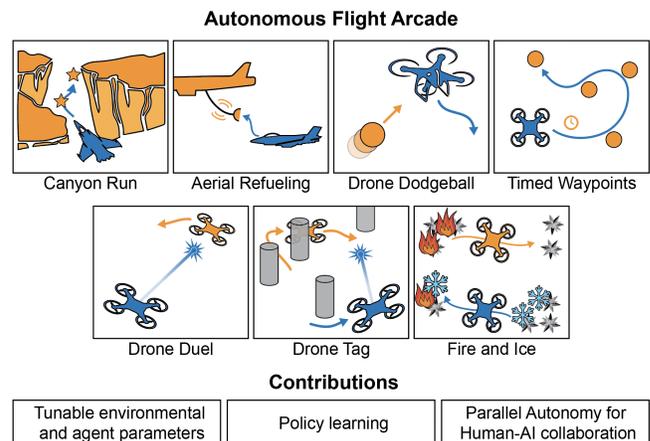


Figure 1: The *Autonomous Flight Arcade* enables learning control for both humans and machines in a diverse set of challenging and flexible aerial environments.

One representative domain where real-world experimentation first requires carefully-designed simulations for training AIs is learned control of aerial vehicles.

In this paper, we introduce the *Autonomous Flight Arcade* (AFA) built around the game engine Unity [7], for learning control of aerial vehicles. The control challenges in the AFA are novel and distinct from those found in other settings. Mastering the tasks requires successfully learning both the low-level details of controlling the aerial vehicles and the high-level planning necessary for achieving complex goals.

Contributions. Our contributions with the AFA are as follows: (1) a suite of challenging and novel flight-domain learning environments; (2) scenarios explicitly designed for both RL and human playability, including a pipeline for human data collection, with tunable environmental and control parameters, allowing for configurations with varying complexity and that emphasize different parts of aerial mission profiles; and (3) the public AFA Challenge. Our framework is integrated with RLLib [10], allowing for scalable

training of RL agents. We also pose the AFA as a challenge problem with a public leaderboard, encouraging submissions of agents in the form of trained models or non-learned control policies.

Related Work. Our work combines the key aspects of both reinforcement learning and simulation environments, namely the challenges of finding a high-quality policy as in other RL environments and the realistic details, including physics and raw control, from simulation environments. Reinforcement learning environments include the Arcade Learning Environment [3, 11], SURREAL [5], Gym Retro [12], dm_control [17], NetHack Learning Environment [9], and Playroom [19] (focusing on natural language tasks), and environments for learning to play zero-sum [4, 13] and cooperative board games [2]. Simulation environments include MuJoCo [18] for rigid-body physics, AirSim [15] for cars and quadcopters, Flightmare [16] for quadcopters, and Gazebo [8] for various vehicles including cars, quadcopters, and fixed-wing aircraft. Beyond model-based simulators, there has been prior work on end-to-end data-driven simulation environments [1, 6].

2 AFA TASKS AND ENVIRONMENTS

Canyon Run is a single-player environment where the agent’s goal is to fly a fixed-wing aircraft through a narrow canyon, below the canyon rim, while hitting waypoints that appear at different heights above the canyon floor.

Aerial Refueling is a single-player environment where the agent is tasked with bringing a jet plane into the vicinity of a tanker plane for a refueling maneuver, prior to running out of fuel.

Drone Dodgeball is a single-player environment where an agent controlling a quadcopter is tasked with station-keeping (i.e., remaining as close as possible to a fixed waypoint) while avoiding oncoming obstacles in the form of dodgeballs.

Timed Waypoints is a single-player environment in which an agent controlling a quadcopter with limited fuel flies above a mountainous terrain with certain “no-fly” (exclusion) zones, where the goal is to reach as many waypoints in 3D space as possible.

Drone Duel is a two-player environment in which two opposing agents try to tag the other agent with a laser.

Drone Tag is a two-player environment in which two opposing agents need to navigate a terrain with obstacles to reach the other drone, and then try to tag it with a laser.

Fire and Ice is a two-player environment in which two opposing agents, the *Fire Drone* and the *Ice Drone*, need to navigate a forest. The Fire Drone has the ability to increase the temperature of trees by hitting them with its laser beam, until they catch on fire. The Ice Drone has the ability to decrease the temperature of trees by hitting them with its laser beam, until they are frozen.

3 BENCHMARK RESULTS

We use realistic but still somewhat stylized models for both the jet and quadcopter. For the jet, we use a custom controller based on the *AeroplaneController* in Unity’s Standard Assets that models lift, velocity-dependent drag, angular drag, and a simple auto-level functionality. For the quadcopter, we use a custom controller that models dynamics including roll, pitch, thrust, and yaw moments, and a restoring torque.

Each environment provides both visual and vector observations. Visual observations are generated through a unified sensor, common to both the jet and quadcopter, consisting of a rectangular array of raycast sensors. The vector observations consist of measurements of vehicle state. We use the proximal policy optimization (PPO) algorithm [14] implemented in RLLib [10] to train benchmark policies. The purpose of these initial benchmarks is not to present agents demonstrating very high performance, but rather to indicate that with a reasonably simple network architecture and using PPO, it is possible for RL agents to learn something in these environments. Given the multiple variations of the environments, we believe that they present a sufficiently high skill ceiling to make for interesting future learning.

4 PUBLIC CHALLENGE PROBLEM

The Autonomous Flight Arcade will have two separate public challenges. In the first challenge, teams will compete to develop the best agent, whether learned, controlled via a planning algorithm, or another approach, such as an ad-hoc heuristic. Teams will have access to compiled executables of the AFA environments for developing their agents. In the second challenge, teams will either compete with or against AI agents in a browser-based format to collect human trajectories. The human trajectories will inform the development of more performant agents for human-machine teams, and will also provide new human baselines.

5 CONCLUSION

We have introduced Autonomous Flight Arcade, a novel suite of challenging learning environments for control of aerial vehicles. By varying environmental parameters, the AFA environments allow for varied levels of difficulty and challenge. While we have introduced some performance baselines in this work, we have also demonstrated the potential for future trained agents to be compared directly with human performance, and thereby serve as a setting for studying human-AI collaboration and competition. We anticipate that these environments will provide fruitful avenues for future research on artificial agents in challenging, realistic situations.

ACKNOWLEDGMENTS

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

The authors acknowledge the MIT Supercloud and Lincoln Laboratory Supercomputing Center for providing high-performance computing resources that have contributed to the research results reported within this paper. The authors are grateful to Ho Chit Siu, Daniel Wrafter, Catherine Johnson, Victor Lopez, Lindsey McEvoy, and the MIT Lincoln Laboratory Serious Games Working Group for their contributions to this work.

REFERENCES

- [1] Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, and Daniela Rus. 2020. Learning Robust Control Policies for End-to-End Autonomous Driving from Data-Driven Simulation. *IEEE Robotics and Automation Letters* 5, 2 (2020), 1143–1150.
- [2] N. Bard, J. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, I. Dunning, S. Mourad, H. Larochelle, M. Bellemare, and M. Bowling. 2020. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence* 280, 103216 (2020).
- [3] M. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *JAIR* 47 (2013), 253–279.
- [4] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890.
- [5] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei. 2018. SURREAL: Open-Source Reinforcement Learning Framework and Robot Manipulation Benchmark. In *Conference on Robot Learning (CoRL)*.
- [6] Winter Guerra, Ezra Tal, Varun Murali, Gilhyun Ryou, and Sertac Karaman. 2019. Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality. *arXiv:1905.11377* (2019).
- [7] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange. 2020. Unity: A General Platform for Intelligent Agents. *arXiv:1809.02627* (2020).
- [8] N. Koenig and A. Howard. 2004. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3. 2149–2154.
- [9] Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. 2020. The NetHack Learning Environment. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*.
- [10] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica. 2018. RLlib: Abstractions for Distributed Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 3053–3062.
- [11] M. Machado, M. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling. 2018. Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents. *JAIR* 61 (2018), 523–562.
- [12] Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. 2018. Gotta Learn Fast: A New Benchmark for Generalization in RL. *arXiv:1804.03720* (2018).
- [13] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.
- [14] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347* (2017).
- [15] S. Shah, D. Dey, C. Lovett, and A. Kapoor. 2017. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*. *arXiv:1705.05065*
- [16] Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. 2020. Flightmare: A Flexible Quadrotor Simulator. (2020).
- [17] Y. Tassa, S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess. 2020. dm_control: Software and Tasks for Continuous Control. *arXiv:2006.12983* (2020).
- [18] E. Todorov, T. Erez, and Y. Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5026–5033.
- [19] T. Ward, A. Bolt, N. Hemmings, S. Carter, M. Sanchez, R. Barreira, S. Noury, K. Anderson, J. Lemmon, J. Coe, P. Trochim, T. Handley, and A. Bolton. 2020. Using Unity to Help Solve Intelligence. *arXiv:2011.09294* (2020).