Reinforcement Learning with X-Plane

MIDN 1/C Matthew Yates and Asst Prof Gavin Taylor
United States Naval Academy
572M Holloway Rd. Stop 9F
Annapolis, MD 21402

Abstract
Reinforcement learning has become a victim of its own success. Problems that used to be considered difficult and interesting are now viewed as easily solved. With some of these domains, it is no longer easy to distinguish whether one RL approach is better than another. To remedy this, new domains to evaluate reinforcement learning algorithms need to be created. Our X-Plane Machine Learning plug-in offers a solution with an easily approached interface to a challenging domain with realistic physics and noise.

Introduction
Reinforcement Learning is one way to pursue autonomy, by leveraging past observed experiences by the agent to learn a useful policy that accomplishes the given task. It is important that research in this area have challenging domains to explore, with realistic physics and noise models, so that the work can be easily applied to real physical systems. We provide such a domain.

In Reinforcement Learning, the domain is modeled as a Markov Decision Process (MDP). We notate the set of states, the set of actions, the transition function, the reward function, and the discount factor as $\mathcal{S}$, $\mathcal{A}$, $P$, $R$, and $\gamma$, respectively.

We are particularly interested in assisting work finding a value function $V(s)$ that maps each state $s \in \mathcal{S}$ to the expected total $\gamma$-discounted reward for the process. In the creation of policies, it can also be helpful to solve for a Q-function $Q(s,a)$, which maps each state-action pair to an expected total $\gamma$-discounted reward.

In large or continuous domains, $V$ and $Q$ can only be approximated. Value function approximation is therefore a robust field of study, needing domains to try new approaches on. We offer an easily-implemented domain which is challenging and realistic, in which the agent can learn to fly a variety of aircraft, in a number of tasks, through realistic weather.

Problem
Evaluation of a RL algorithm requires a domain where a learned policy can be analyzed as it tries to achieve a final goal. Canonical examples of these domains are the mountain car problem (Sutton and Barto 1998), the inverted pendulum problem (Wang, Tanaka, and Griffin 1996), and the bicycle problem (Randløv and Alstrøm 1998).

These traditional problems for evaluating policies can separate the poor approaches from the good. However, the field has advanced enough that these problems no longer offer an ability to separate the truly great and innovative learning algorithms from those that are merely good. Newer and better approaches to RL do not have enough room to improve upon the success of the previous generation of algorithms to show their merits.
In the next section, we propose an alternative domain that addresses this problem.

**X-Plane Domain**

As a new domain we offer the X-Plane Flight Simulator. To use X-Plane as a domain we created a simple XML API designed for Reinforcement Learning that can set flight conditions, sample data, and follow a policy implied by a learned Q-function approximation.

X-Plane closely models the real world with 3813 “datarefs” that can be read and set while X-Plane is running. With these datarefs every single control surface’s position can be both observed and set and every value that the plane has a sensor for can be read.

While we demonstrate training X-Plane to take off from a runway, the plug-in is not limited to take off. For example, a researcher may have the plane start in the air at a desired speed and altitude for learning a standard rate turn, or rescuing itself from a nose dive or spin. Further, whenever a RL algorithm can solve one problem in X-Plane a researcher can always change the plane the simulator is using or add uncertainty stemming from weather.

Although the X-Plane domain is still an abstraction from the real world it provides a very accurate simulation, as the realism in the simulation allows the researcher to experiment with lifelike noise without the complications of physical systems.

**Proof of Concept**

To demonstrate the merits of using X-Plane as a domain for evaluation of Reinforcement Learning we created an agent trained with Least-Squares Policy Iteration (LSPI) ([Lagoudakis and Parr 2003](http://www.usna.edu/Users/cs/taylor/xplane)) to take off a Boeing 747. Actions were made every second. We defined four features for each state: the altitude, the ground speed, the pitch, and the position of the elevators. We defined our action space by discretizing the space of elevator settings into five settings, ranging from completely up to completely down.

Samples were collected by starting motionless on the runway with engines on full, and performing random actions until a crash. 32 such runs were performed in the sampling stage.

Using LSPI, we were able to approximate the Q-functions well enough that, by choosing the action associated with the Q-function at the current state with highest value, the aircraft was able to autonomously take off.

Videos of both random sampling and the successful policy are visible on our web page ([http://www.usna.edu/Users/cs/taylor/xplane](http://www.usna.edu/Users/cs/taylor/xplane)).

**Conclusion**

Reinforcement learning needed more difficult domains to challenge modern techniques. We created a new, more challenging domain using a modified X-Plane flight simulator that can be used to properly evaluate RL algorithms. To prepare X-Plane for use as a domain for testing Reinforcement Learning approaches we created an easily-installed and easily-adapted plug-in that collected data, set up flights, and input commands without a person in the loop.

X-Plane is a usable domain for reinforcement learning and will be in the future useful to researchers to give their algorithms environments with extensive noise and large action spaces. This domain also moves RL research closer to the development of real world autonomy.

**References**


