Using Traffic Data Management to Enhance Safety of a Reinforcement Learning Agent
Project Proposal

Evana Gizzi[1], David Zabner[1], and Vincent Houston[2]

[1] Tufts University, Medford MA 02155, USA
[2] NASA Langley Research Center, Hampton VA 23666, USA

1 Project Overview

In this project, we are exploring the ability of a basic reinforcement learning (RL) agent to land and aircraft on its approach to the runway. Specifically, we want to see how the agent preforms with consideration of other aircraft in the airfield, as a way to measure its ability to regard airfield safety measures on its landing. Furthermore, we will be integrating data from an intelligent traffic data management system to see if the input helps the intelligent agent with its overall success of landing, and optimizes its safety while doing so.

As such, we have the following project aims:

1. Can we control an aircraft with high level commands?
2. Can we use high level commands to approach a runway?
3. Can we fly a safer approach with a traffic data management system?

The Traffic Data Manager (TDM) system was developed under the advise of our coauthor, Vincent Houston, at NASA Langley Research Center, as a supervised learning system which is able to determine whether aircraft's in an airfield are relevant enough to a flyer to be considered dangerous to a real-time trajectory. The data used to train the learning algorithm was human labelled from subject matter experts, namely, commercial aviation pilots. TDM has been benchmarked for its performance against labelled datasets through comparison and validation studies, but has yet to be observed in an live, online learning setting. Thus, it has never been observed in a human-in-the-loop setting, let alone a fully autonomous environment. It is our goal to observe how TDM enhances autonomous performance, and also to observe how the safety measure of our RL system is effected by the integration of this data.

In the following, we describe our application domain and the tools available to us for this research effort.

1.1 Domain

In this research, we will be exploring the commercial aviation domain. Specifically, we will be using a scenario in which an intelligent system (built from our

RL agent) is trying to land an aircraft. We will proceed in two stages: First we will show that an RL agent is capable of landing an airplane and second, we will show that an RL agent is able to choose a safer flight path when given information about other aircraft in the area. Landing takes place in two main stages, the approach, which is the larger portion of the landing bringing the plane from cruising altitude to 100 feet in altitude, and the flare, which is the final landing portion bringing the plane from 100 feet to the ground. The flare is, in many ways, the more complicated portion of this as the flight dynamics at low altitude include ground effect, heavy crosswinds, etc. Because the flare is more complex and is not dependent on other aircraft we will likely ignore it, ending our simulations once our agent has brought the plane to 100 feet above the runway.
Scenario Our scenario will consist of the descent of a Boeing 757 into the Reno International Airport, which sits at 4450 ft above sea level. The simulation will start at 8000 ft above sea level, and will attempt to navigate the aircraft safely to an altitude of 4550 ft above sea level. The simulation will start facing the runway head on. This scenario is subject to change based on compatibility with TDM scenario data, based on scope, and based on other experimental factors.

1.2 Tools

VISTAS: We will be working with a rapid proto-typing simulation environment located at NASA Langley Research Center in Hampton VA (VISTAS simulator). We will have a copy of an interface to the simulator that we will be using which provides close to full range capabilities of the simulator (provides enough for our project). The data to be used to define our state space will come from the frames recorded from the VISTAS. The fields in section 1 of the appendix (Section 4.1) are available to us for defining our states (populated with sample data).

Alternatives: An alternative simulation environment that we are considering is X-plane has been used by many researchers in the past [2] [4] [3] [1]. The reason why we have this consideration is because of potential issues with releasing data/software from NASA, so we want to make sure we have an alternative option. We plan on building our system in a way that is agnostice of the simulation environment, with two 'plug-in' simulation interface modules that we will build in house. We already have a proxy to the VISTAS simulator that was built by Evana Gizzi in 2017. This simulator has been successfully used by many previous investigators as a Reinforcement Learning environment.

2 Background and Related Work

A variety of work has already been done showing that it is possible, through a variety of different RL methods, to safely pilot a plane through take-off, landing, cruise and navigational aspects [4] [3] [1] [2]. However, no-one we have found has

given the RL agent control over all aspects of flight including trajectory management based on environmental information including data about other aircraft in the airfield. This is where our work will be novel and, hopefully, show results indicating that RL can be used to increase flight safety as well as suggesting that TDM is advantageous for human pilots by allowing them to take in less information.
It is also worth noting that our work will be novel in the sense that the majority of past work we found on RL and flight used apprenticeship learning and "expert instructors". Although, we may be able to provide a certain amount of instruction to the algorithm, our lack of access to "experts" will present an intersting, important, and novel challenge in the RL-Flight literature.

3 Problem Formulation

We begin the following section with a description of an application specific problem formulation at a theoretical level. We describe how we formulate our problem within the context of a general reinforcement learning problem, which includes a *state space*, *action space*, *reward function*, and *goal state*. We proceed this formulation with a description of our implementation plan.

3.1 Theoretical Problem Formulation

State Space: We define our state space S as a set of states $S_1, S_2, \ldots S_n \in S$ as a collection of values corresponding to the following:
$\{LongitudeDegree, LatitutdeDegree, RadarAltitudeFeet, Pitch, Roll, Yaw,$
$Airspeed, Throttle, AileronPosition\}$.

Action Space: We define our action space as a direction vector v and a discrete trajectory style chosen from the following set: $\{SPEED\_UP, SLOW\_DOWN\}$. This small action set will have to feed into a system capable of translating it into low level airplane actions like $\{IncreaseThrottle, MoveElevator,$
$MoveLeftAileron, MoveRightAileron, MoveRudder\}$.

Reward Function: The agent will receive a small negative reward for each time-step and a large negative reward for "bad-orientations" and for going over a g-limit and a large positive reward for matching, to within some delta, the goal orientation and speed which will also end the episode.We don't know a lot about what a bad orientation really is so we will just assume that our plane should never exceed a 60° angle in pitch or roll.

For the TDM portion of our project we will engineer our reward function with regard for the safety measures for the landing phase of flight in commercial aviation. We scope this to the goal of not colliding with, or coming within a predefined proximity of other aircraft in the airfield. We will access the airfield information through TDM input data, and use the TDM technology to identify aircraft as either *relevant*, *maybe relevant* or *not relevant*.

Goals: The overall goal of the agent is to get the aircraft below 100 ft and to optimize the safety function overall. There is still some uncertainty regarding what precisely makes a flight “safe”. However, within this context we have two resources. The first, is access to the National Aeronautic and Space Administration who have offered to provide feedback and guidance. We hope to get precise definitions from them about what constitutes a safe flight. The second, is that success in the face of ambiguity would be further proof of the utility of RL in the flight context. If we can show that under multiple different safety regimes, RL maximizes safety, than we will have shown that RL is very widely applicable to maximizing safety in an aviation context.

3.2 Implementation Plan

In the following, we describe our implementation plan in the context of our 3 project aims.

1. Can we control an aircraft with high level commands? The first goal is going to be to show that using reinforcement learning, we can develop an agent capable of carrying out specific high level commands in relation to changing the orientation and speed of the aircraft. These commands will consist of combinations of changes in the pitch, roll, and yaw of the aircraft as well as the airspeed. These commands will then serve as a basis for the next to sub-goals. All of the actions taken by this agent will keep the plane in a safe (mostly upright) orientation, hopefully, be g-limited for ”customer comfort” although the g-limiting is very much a secondary goal. The input to this agent will consist of information about the orientation of the planes control surfaces, the current pitch, roll, yaw, airspeed, and a g-force vector along with the distance from the goal pitch, yaw, roll, and airspeed. The agents action space will be either... figure out if we can give the planes ”directional/joystick” controls or if we need to independently control the throttle and each of the control surfaces.

2. Can we use high level commands to approach a runway? The second goal will be to use the learning from part 1 as options [5] to accomplish the goal of approaching the runway of an airport. The end goal here is to reach a certain region of space at a certain speed and orientation. Some type of training regimen, like increasing the distance from the goal over episodes, is also likely to help.

3. Can we fly a safer approach with TDM? The final goal will be to increase the safety of the approach and in the process compare the utility of using raw data about the location, direction, etc. of the other aircraft in the airfield against that of TDM selected data in increasing approach safety.

4 Evaluation and Expected Outcomes

In this work, we are looking to evaluate the performance of our system overall, along with benchmarking the performance on a more granular level. In order to accomplish the overall evaluation, we will observe our output in the context of our research questions. In order to evaluate the more granular performance of our system, we will develop a set of benchmarking measures.

4.1 Research Questions and Expected Outcomes

The main research question that we are attempting to answer with this proof of concept model is whether TDM data helps increase the safety of autonomous air systems. In past work, the effectiveness of TDM was evaluated against labelled datasets to evaluate its accuracy, but has yet to be evaluated for its effectiveness in live, online scenarios. We will evaluate its effectiveness by measures how safety is effected by the integration of TDM data in an RL agent by comparing output of our safety function in different trajectory trials.

4.2 Benchmarking and Evaluation Techniques

We will evaluate the lower level performance of our system by looking at multiple factors, some of which are still to be determined. Ultimately the most important factors will be safety, fuel usage, and speed of task completion in that order. Interestingly, speed is the easiest of the three to understand, specify and measure, and safety is the hardest but we are hoping that the evaluation of safety is more of a long term goal.

References

1. Haitham Baomar and Peter J. Bentley. An intelligent autopilot system that learns piloting skills from human pilots by imitation. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016. This paper covers the building of a fully autonomous autopilot system. It used a very short learning period combined with a very small amount of apprenticeship learning to get very impressive results.
2. Haitham Baomar and Peter J. Bentley. Autonomous navigation and landing of large jets using artificial neural networks and learning by imitation, 2017. This paper covers the building of a fully autonomous autopilot system for landing large aircraft. It used a very short learning period combined with a very small amount of apprenticeship learning to get very impressive results.
3. Oren Hazi. Final approach an automated landing system for the x-plane flight simulator. This unpublished paper explores the problem of landing a plane in the X-Plane flight simulator with RL and Imitation Learning.
4. Eduardo F. Morales and Claude Sammut. Learning to fly by combining reinforcement learning with behavioural cloning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML ’04, pages 76–, New York, NY, USA, 2004. ACM. This paper uses “Relational Representations” of the state action space for approximating the state action space in flying an airplane.

5. Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, August 1999. This paper explores “Options”, which may be an effective approach for speeding up learning by setting short range goals in the landing process.