Air Traffic Control with Reinforcement Learning

Marut Priyadarshi[1] and Meet Maratha[2]

[1]marut19@iiserb.ac.in — 19191
[2]meet19@iiserb.ac.in — 19190

May 6, 2022

1 Motivation

Since the drastic reduction of cost of air travel over the past decades, the volume of aeroplanes in airports and in the skies, at any given moment, has increased exponentially. Coordinating all these planes landing, taking off, and just in the general airspace is no mean feat, where even a single human error, or a lapse in judgement could result in the loss of hundreds, if not thousands, of precious human lives. So here, we aim to replace the human part of this process with AI trained with reinforcement learning on a game that simulates the Air Traffic Control. The agent, once properly trained, has the potential to be much more adept at this job than a human can, and without the possibilities of human error.

2 Problem Statement

In the ideal case, a perfectly trained AI would handle all Air Traffic Control, giving directions, and commands to ensure that the whole air travel system keeps working smoothly and without incident. But in reality, most of it is controlled by humans, aided by technologies, but the important decisions are all man made and are subject to the risk associated with anything that involves humans. Errors in this case are very costly, both in terms of life and resources. A single crash can lead to the loss of lives and disrupt a huge part of the air travel network.

Our goal here is to train an AI model that is capable of doing this job on its own. For training it, we have used a game that simulates the Air traffic control system and has the goal of redirecting planes to prevent any collisions and make the planes safely reach their destinations. We trained our agent on this game so that it can do all those tasks autonomously and efficiently without any need for human interference.

3 Approach Taken

First of all, we have to modify the original game, as it was made to be played by a human player. We have to change it so that the agent can directly affect the game and choose actions based on the policy. We have to change the menu of the game to accept the different parameters that the agent and the game will need to run, such as: number of planes, number of destinations, the exploration factor etc.

Then we have to define the environment, the collision criteria ($d < 50$, Figure 1), the rewards from different states and the set of actions. We are using two criteria for rewards, for maximizing the distance between two planes, and minimizing the total distance to the destination:

$$Intruder\_Reward = \frac{-(radius^2 - closest\_distance^2)}{radius^2/500}$$

$$Distance\_Reward = 100 - Distance\_ToGo$$

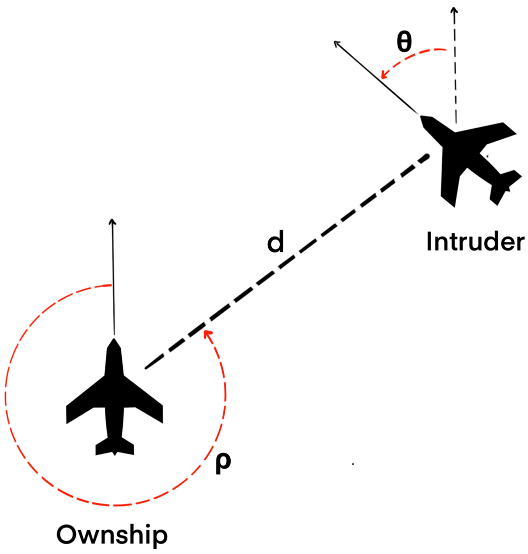

Figure 1: Description of state space:

The theory is that if two planes come within 50 units of within each other, they take one action from a set of actions (Table 1) depending on the angle between the two planes as shown in the Figure 1. The rewards are direct proportional to the closest distance between the planes, and inversely proportional to the total distance to the destination left after the action. This continues until the planes are outside the danger radius.

Then we come to the training part. We have used Epsilon Greedy and SARSA algorithms to train our agent over the course of 600 episodes.

Table 1: List of Actions

Action	Description
N	Maintain Course
HL	Hard Left by 72°
HR	Hard Right by 72°
ML	Medium Left by 36°
MR	Medium Right by 36°

4 Results

We were able to:
• Able to create an agent that is capable of playing the game without any human aid

• Successfully created an environment that simulates the 2d version of ATC and defined all the rewards, states and actions for it.
• Trained the agent over the course of 600 episodes using Epsilon Greedy and SARSA

Thus, we were able to achieve our primary goals of creating an agent that is able to do the tasks of Air Traffic Control to prevent collisions in mid air or during take off or landing.

Our current agent is still quite primitive when compared to the level of details that exist in real life Air Traffic Control systems. So future works can focus on taking this to the third dimension, and adding physics based restrictions because an airplane cannot turn or respond to commands instantaneously. More efficient algorithms can also be focused on, to create a better performing agent.