International Journal of Engineering Technology and ManagementSciences
Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

Flight AI using Reinforcement Learning

Poonam Sharma[1], Ashima Narang

[1]*Assistant Professor, Amity University, Haryana*
[2]*Assistant Professor, Amity University, Haryana*

[1]psharma1@ggn.amity.edu

***Abstract*—** This paper will be featuring an aircraft with fully featured physics simulated in unity engine. The agent or the AI will be created for possessing and controlling the aircraft so as to navigate it in the 3D world space environment provided. The agent will have to consider the different physical dynamics applied on a real-world aircraft and based on these parameters it will have to create an effective piloting for the aircraft. Various other aspects such as engine dynamics and the fuel parameters will also be considered for an effective training environment.

1.Introduction

Machine learning has been utilized for various aspects for the development and growth of various fields across the world. After the development of deep learning technology, various problems are easily resolved via the remedies provided by machine learning/deep learning algorithms and techniques. From stock predictions to sentiment analysis, machine learning has been utilized in various applications for better results. One such problem is the creation of an efficient yet simple artificial intelligence that can help control a complex piece of engineering.
We decided to test our skills and boost our knowledge by putting together such a project in which we can test the extremes of an AI via presenting it with the possession of an advance piece of human engineering mastery.

2. Background Study

Aviation is widely regarded as the safest means of transport. Continuous advancements from the aviation industry alongside with strict international rules make that the number of incidents per year are still declining, whilst the total number of flights are increasing [1-4]. The introduction of Flight Control Systems (FCS) has enhanced this safety by adding closed loop stability, putting boundaries on pilots' inputs, and reducing the pilot's workload. FCSs have been designed using linear control theory for many years, with satisfactory results. Nonetheless, linear control theory suffers from performance degradation due to non-linearities, uncertainties in the model, and faults or damage taken by the aircraft. Furthermore, the design of FCSs using linear control theory is costly due to required gain scheduling because of the range of dynamics in the operating range of the aircraft and the need for an accurate model which might not be readily available. The performance degradation can also lead to safety issues when damage occurs. With the rise of autonomous systems this need for adaptive control and safety becomes even more apparent.
To combat the performance degradation a lot of research has been performed and proven on aircraft. For example, Doyle, Lenz, and Packard have shown H∞ loop shaping in combination with μ-synthesis, whilst Kulcsár has shown how a Linear Quadratic Regulator (LQR). Both

Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

allows for the shaping of the controller such that the robustness and performances are balanced. However, both are mathematically complex linear methods and therefore less effective as the system becomes more complex and non-linear.

”Reinforcement learning is learning what to do, how to map situations to actions, so as to maximize a numerical reward signal.” Actions may have effect not only on the immediate reward, but also on future rewards. An algorithm, or agent, must learn from its own experience and is not told what action to take or what reward can be expected. In this research the FCS takes the role of agent and must choose which action to take such that the aircraft behaves as desired. The acted will influence the environment, consisting of the full aircraft dynamics, and in turn the agent receives the states and reward signal as feedback to close the loop. From these states the agent will choose a new action again. From experience the agent must learn what actions is best given a certain state. This is done by maximizing the cumulative rewards received, where the rewards signal can be seen as a rating for the state the agent is in. Using a reward signal to reach a capable controller is a key concept of reinforcement learning. The reward signal should therefore be designed such that it guides the agent towards the desired controller. For example, if we want an agent to fly an aircraft and follow a certain trajectory, we can design the reward signal such that it receives a reward of 1 when it follows the trajectory and a reward of 1 minus the tracking error when the aircraft deviates. In the long run the agent will receive the maximum reward when it follows the trajectory precisely and will learn to do so [5-6].

One of the big challenges in reinforcement learning is the amount of simulation that needs to be performed for the agent to converge to a solution. All state-action pairs must be visited many times before the solution converges. As the state space and action space increases the amount of simulation required also increases. To use continuous state- and action-spaces and allow for faster convergence faster convergence an approximation of the policy function can be used, or so-called policy gradient method. Different types of approximations can be used, but in this research each function is approximated using a neural network that maps the respective input to the respective output.

The scope for simulation will be enclosed nearly in:

• Thrust (Variations due to altitude and air rarefaction)

• Lift and drag coefficient (variations due to angle of attack, including stall and variations due to Mach number impact on lift and drag)

• Drag (due to air density variation with altitude and due to the angle of attack variations)

• Lift (Flap usage to increase lift)

• Fuel consumption (specific fuel consumption estimation and aircraft mass variation due to consumed fuel.

The aim of the aircraft model is to compute the acceleration, speed, and position (which we will be calling dynamics in the rest of the project) of the aircraft for any given timestep based on conditions of the previous timestep. Throughout this study, we will be using the ground frame of reference.

International Journal of Engineering Technology and ManagementSciences
Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

3. Methodology

3.1 Formulation

How do aircrafts fly?
In order to create a flight model, we first needed to grasp the basic idea of an aircraft and its physic-based dynamic. There are four forces that affects an aircraft.
The two horizontal forces are:
• Thrust: Force created by its propeller(s) or reactor(s) pushing the aircraft forward
• Drag: Force created by the air resistance and thus opposed to the aircraft's movement

The two vertical forces are:
• Weight: the weight of the aircraft

• Lift: The force allowing the aircraft to fly.

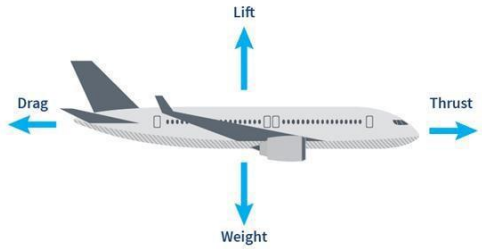

Fig. 1: Forces on an Aircraft Flight

3.2 Aerodynamic Modelling
3.2.1 Mathematical Formulation
To compute the dynamic of the plane at a given time step in the frame of reference of the ground we will use the following relations:
Eqn

$$\overrightarrow{\Delta V} = \vec{a}.\Delta t \quad (1)$$
$$\vec{V}(t+1) = \vec{V}(t) + \overrightarrow{\Delta V}$$

- *V being the velocity* $(m.s^{-1})$
- *a the acceleration* $(m.s^{-2})$ *and*
- Δt *the time interval between two frames of our model* (s)

We now have a way to calculate velocity and position based on acceleration. Let's now calculate the acceleration. To do this we will be using Newton's second law, in other words, the sum of the forces applied to an object is equal to its mass times its acceleration [7] . So, to compute our

Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

plane's acceleration we have to compute the sum of the forces applied to it. As presented before we have weight, thrust, drag, and lift. Let's take a look at each one individually.

3.2.2 Weight Calculations

It's the force due to Earth's gravity reacting with the plane's mass.

$$\vec{W} = m.\vec{g}$$

- m is the plane's mass (kg) $for\ an\ A320\ we\ have\ m = 73.5\ t = 73{,}500\ kg$
- $\vec{g}$ is the gravitational acceleration vector (directed towards the center of the earth) $(m.s^{-2}) = 9.81\ m.s^{-2}$

3.2.3 Drag and Lift Calculations

Drag is basically the air resisting the plane's movement. Lift is the force created by the difference between speeds above and beneath the wing allowing to compensate for the weight and therefore gain altitude. Even though their impact on flight is totally different, they are both aerodynamic forces and are obtained through a similar formula:

$$F = \frac{1}{2}.\rho.C.S.V^2$$

- F : $Resulting\ force, either\ drag\ or\ lift$ (N)
- ρ : $air\ density\ (kg.m^{-3})$
- C : $drag\ or\ lift\ coefficient\ (no\ unit)$
- S : $reference\ surface\ (m^2)$
- V : $velocity\ (m.s^{-1})$

3.2.4 Reference Surfaces

The reference surfaces, needed to compute drag and lift, are the surfaces orthogonal to the direction of the force we are interested in. For drag the reference surface is the one opposed to the relative wind, whereas for lift the reference surface is the one parallel to the relative wind. The surfaces we are interested in are the one relative to the relative wind [8].

We will, therefore, be using the relative wind's frame of reference (x'z'). It is obtained by rotating our initial frame of reference by the value of the slope.

We must calculate the projection of the front and wings surfaces to the relative wind's frame of reference.

International Journal of Engineering Technology and ManagementSciences
Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

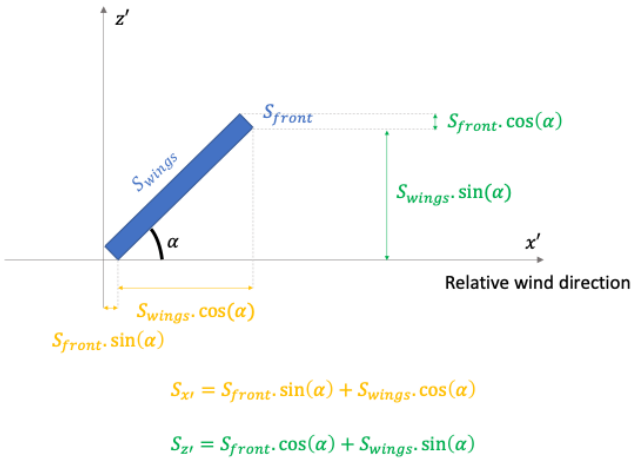

Fig 2: Resolution of Reference Surfaces

We thus get those equations for the projected surfaces along our relative wind axes x' and z'.

3.3 Agent Formulation

3.3.1 Architecture

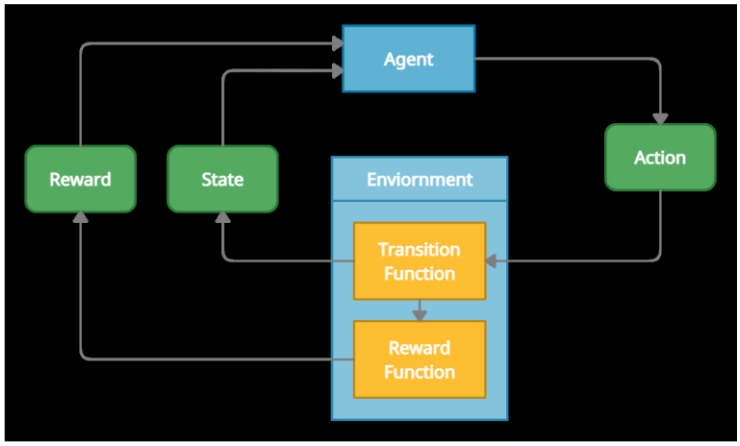

Fig 4: Agent Architecture

The Agent initializes in an initial state and then performs an action based on the observations.

• The observations are calculated via the influence of actions on environment by the transition function.

• The transition function then interprets the new state and forwards this as feed for reward function.

International Journal of Engineering Technology and ManagementSciences
Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

• The reward function then modifies it so that reward can be maximised out of the actions.

• The Agent then again gets feed this data and interprets and takes next action. \

3.3.2 Using a neural network to represent the autopilot

To model the agent (the pilot), we then built a simple feedforward neural network using TensorFlow. As inputs, this model took the difference of the airplane's desired altitude and its actual altitude Δh and the current pitch angle θ. The outputs were, as mentioned before, one of three allowed actions, supposedly such as: (1) increase pitch angle, (2) decrease pitch angle, or (3) maintain current pitch angle. Our network looked like this:

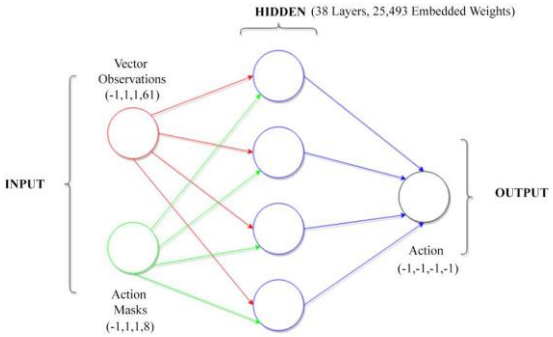

Fig 5: Agent Neural Network

Our basic neural network with a single hidden layer, which we used to make decisions about how to adjust the pitch of our model aircraft. Our network had 2 inputs, 38 hidden layers, and 19 an output representing the action. We experimented with different numbers of neurons in the hidden layer.

4. Implementation

4.1 Aircraft Design

The aircraft design is accomplished via 3D modelling in Blender 3D. It is modelled after the Embraer EMB-120 with some design modifications preserving the original aerodynamic structure of the craft. The aircraft is broken in two separate models – one for fuselage and another for wings with propellers and landing gear as shown in fig 6.

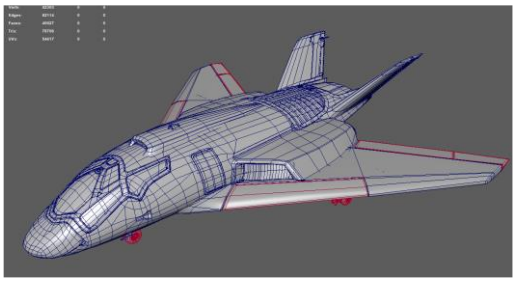

Fig. 6: Aircraft Model

International Journal of Engineering Technology and ManagementSciences
Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

4.2 Results

Given below are the different plots obtained via the tensor board while training our agents. We used visual interpretation for evaluation as it will be much easier to evaluate the precision of agent and verify them.

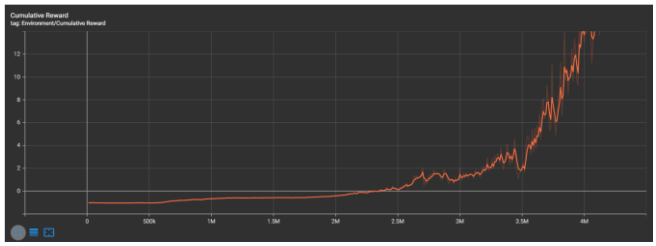

Fig. 6: Visualization of Cumulative Reward

Fig. 6 represents the nature of cumulative reward with respect to step. As it can be clearly perceived that the cumulative reward for the agent increases with the increasing steps.

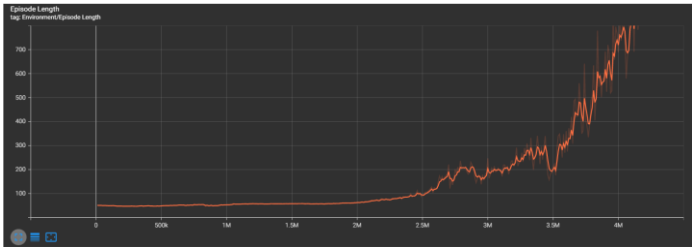

Fig. 7: Visualization of Episode Length

Fig 7 illustrates the nature of episode in relative to the steps elapsed. Again the episode length goes on increasing with the increase in steps.

5. Conclusion

This paper contains the motivation, design patterns, detailed implementation, and testing processes for the creation of an agent capable of controlling a physic-based aircraft. After having completed the analysis presented in this thesis report, we can conclude that the model has performed well and has proven to be quite efficient for the task. This experiment showed us that reinforcement learning can be used to create a self- automation Flight system using AI. The agent can be easily trained for just a simple situation like this one then can be challenged to work under different circumstances, as it can easily adapt and learn to manoeuvre in new environments as well.

Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0010 ISSN: 2581-4621

References

[1] Mannucci, T. et al. “Safe Exploration Algorithms for Reinforcement Learning Controllers”. In: IEEE Transactions on Neural Networks and Learning Systems 29.4 (2018), pp. 1069–1081.
[2] Minsky, M. L. Theory of neural-analog reinforcement systems and its application to the brain model problem. Princeton University., 1954.
[3] Flight Controller Synthesis Via Deep Reinforcement Learning, William Koch url: https://arxiv.org/abs/1909.06493
[4] https://towardsdatascience.com/how-i-taught-a-aircraft-to-fly-using-rl-c170a152b771.
[5] Tensorflow: An Open-Source Machine Learning Framework for Everyone. https://github.com/tensorflow/tensorflow/
[6] https://www.mdpi.com/2226-4310/8/1/18/pdf
[7]Flight Controller Synthesis Via Deep Reinforcement Learning by William Koch https://www.researchgate.net/publication/335854955_Flight_Controller_Synthesis_Via_Deep_Reinforcement_Learning
[8]https://medium.com/paladin-ai/teaching-machines-to-fly-themselves-using-artificialintelligence-b70e208574c4