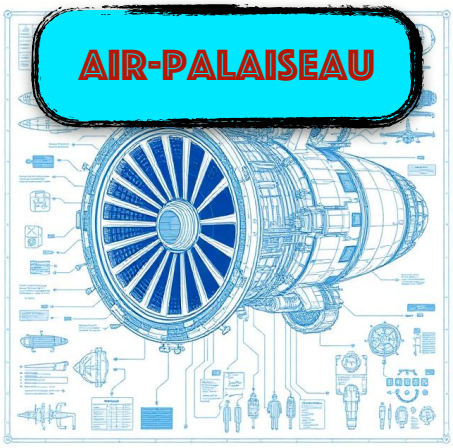
The RL Aircraft Challenge



You are a freshly arrived research scientist of AirPalaiseau, a new airline company that aims to be competitive by bringing AI-driven solutions. The airline makes its money from revenue, luckily all its flights are fully booked. But there are operational costs associated, namely in terms of fuel costs and maintenance that are to be optimized.

Your job is to craft a RL agent policy that seeks to maximize the operational revenues of the planes' engines, while reducing operational costs. You will focus on the engine, and to help you out, there is a RL environment you can leverage to simulate its navigation. You namely have seven sensors measurements that are collected on each flight that act as your observation space. The engine degrades through time, and when needed you can perform a 'repair' action which brings health back to the engine. Each time a repair is made, the engine works better and will use less fuel, bringing your revenue higher! But each repair is costly, therefore you should learn when to precisely apply them. Of course, there is a risk of an engine breakdown which is a catastrophic failure, and brings a hefty penalty to AirPalaiseau, which makes the engine irreplaceable as it is damaged beyond repair. Furthermore safety first! To avoid engine destruction, you can also choose to sell the engine for a price, but the most the engine is used, the less money you'll get from spare parts.

Thus your job is to craft a policy to minimize operation risks, while maximizing operational revenues. As an airline, there are many regulatory concerns and thus you should ideally be able to explain how your agent operates and why it makes certain choices.

Each time a maintenance is made, it does not totally restore the engine's health, and it shortens its lifespan a bit more each time as some components are not replaceable.

Your observation space consists of 9 entries: 7 sensor measurements, the operating condition of the flight, and the timestep.

You can do three actions with their associated rewards:
- 0: do nothing, which rewards you of the profit of that flight based on the health status of the engine
- 1: repair, with costs associated
- 2: sell, retire the engine, get money from the spare parts and end the episode

All episodes begin from a pristine state, out of the factory engine. Small discrepancies exist between engines due to manufacturing, but the real changes will come during the engine's lifetime.

For any question: thil@lix.polytechnique.fr, use 'RLAircraft' in the object field of your email.

The gym environment

The source code can be found here. Additional features might be rolled out in time.

Each time step brings an observation of 9 sensors. You can place actions every 10 time steps, and the action is performed on the first time step returned.

GitHub

Thus, when you start an episode you are at step 0, when your first observation return a vector of dimensions [10, 9] where 10 represents the amount of time spent. Each time you call env.step(.), time advances of 10 timesteps. The choice is yours to decide how you define the input of your model: full batch, or step by step.

There is also a vectorized environment that is proposed which run multiple environments in a vectorized form. If you set up 4 environments, then the returned observations will be of dimensions [40, 9]. You can check the 'info' field that is return alongside each step, defined as:

observation, reward, terminated, truncated, info = env.step(action)

Terminated is a boolean indicated if the episode has ended (True). An episode ends either by engine failure or when a sale has been performed for a reward. Truncated won't be needed for now. Info is a dictionary containing the current step of the episode,

Evaluation:

We'll have a live leaderboard where we can take the last 100 episodes practiced and review the average cumulative reward over these episodes. You will also have to beat a classic baseline used in many airlines, which is a time-based, threshold-based conservative maintenance policy. Furthermore, providing safety guarantees to your solution will be positively looked upon.