DOI: 10.13009/EUCASS2023-633
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

A Reinforcement Learning Approach to Conflict Resolution of Climate Optimal Trajectories

Fateme Baneshi⋆† *and Manuel Soler*⋆
⋆*Department of Aerospace Engineering, Universidad Carlos III de Madrid, 28911, Leganes, Spain*
fbaneshi@pa.uc3m.es · masolera@ing.uc3m.es
†Corresponding author

Abstract

This paper presents an automated decision-making framework based on multi-agent reinforcement learning to address aircraft conflict resolution in high-density traffic scenarios. The proposed strategy leverages a deep deterministic policy gradient algorithm to provide speed advisory to operating aircraft with the aim of minimizing the number of conflicts. To address the non-stationary environment challenge, a centralized learning, decentralized execution scheme is implemented. In the training process, the agents make use of information from other aircraft, but this information is not utilized during the testing phase. The effectiveness of the proposed approach is validated by analyzing various sets of climate-optimal trajectories, which pose critical safety issues due to the uneven distribution of flights within the airspace. The results demonstrate that the proposed framework effectively resolves a high number of conflicts that arise as a result of adopting climatically-optimal trajectories.

1. Introduction

In light of the expansion of global air traffic, addressing the environmental responsibilities of the aviation industry has become a critical challenge.[10] In particular, aviation contributes to global warming by emitting CO_2 and non-CO_2 species. Despite CO_2 being the most visible contributor to climate change, non-CO_2 emissions have been shown to have two times higher impact on climate change compared to CO_2 emissions alone.[14] In contrast to CO_2, the climate impact of non-CO_2 emissions depends on the atmospheric location and time of the emissions.[26] Thus, they can be mitigated by determining more efficient maneuvers to avoid areas of airspace sensitive to aircraft emission, called climate hotspot areas.[14] However, planning climate-friendly routes for each individual flight is not the ultimate solution to this problem.[2] Ignoring the interactions between flights in trajectory planning leads to an incomplete understanding of the potential for mitigating climate impact. The climate-optimal trajectories tend to avoid climate-sensitive areas by rerouting or changing their altitude. Such tendencies alter the traffic distribution by evacuating sectors that include climate hotspots and increasing the density around neighboring ones.[2] The rise in traffic density around climate hotspots significantly amplifies complexity, particularly the number of conflicts, thereby posing substantial threats to air traffic safety. Such possible traffic safety degradation calls for the development of an efficient strategy at the network level that, on the one hand, mitigates the climate impact of aviation by trajectory planning and, on the other, guarantees flight safety and efficiency.

In the current air traffic system, the Air Traffic Control (ATC) is the centralized point responsible for making tactical decisions to maintain the safe distance between aircraft. In this respect, centralized conflict resolution has been extensively studied by researchers. In these studies, different methods were utilized as optimization techniques, such as heuristic[7] and exact methods.[18] Exact algorithms, such as mathematical programming, are primarily used to seek exact (local or, if possible, global) optimal solutions.[18] However, such approaches often require long computational times, making them impractical for real-time applications.[23] The heuristic methods have been employed in some studies to find conflict-free maneuvers in a computationally more efficient manner (though yielding approximate optimality).[12] While heuristic approaches show acceptable performance in solving conflict resolution problems, their execution time scales up with the number of aircraft involved. Additionally, they need to be performed from scratch whenever the scenario changes or new sets of trajectories are received. This complexity is further increased when adopting climate-optimal routes (due to the difficulty of finding conflict-free maneuvers for each set of trajectories as certain areas become crowded due to the avoidance of climate hotspots (e.g., see[2] for a study on conflict resolution of climate-optimal trajectories using the simulated annealing). In this regard, it becomes imperative to develop an autonomous system for separation assurance with fast execution time to meet the real-time demands of air traffic. Given the complexity and

DOI: 10.13009/EUCASS2023-633

dynamic nature of air traffic, a distributed system offers significant advantages by allowing for the reallocation of the conflict resolution process to individual flights.[23] Such system should incorporate self-evolving models to efficiently adapt and learn from new information, facilitating timely and accurate decision-making, even in unseen and large-scale scenarios.[27]

Deep Reinforcement Learning (DRL) is a promising approach to building the foundation of autonomous systems, which has shown good potential in solving sequential decision-making processes previously done by humans.[1] For instance, thanks to the advances of RL algorithms, an artificially intelligent model called "alpha Go" defeated the human world champion in the game "Go".[24] The RL methods also have been used for end-to-end autonomous cars.[6] Due to the capability of DRL methods to solve complex problems, it has also been utilized to build envisioned automated systems for air traffic control applications.[21] For instance, DRL algorithms have been investigated for conflict resolution problems in both en-route and urban areas.[15,19,22] The work conducted by Pham et al.[19] propose a conflict resolution strategy using the Deep Deterministic Policy Gradient method. A circular area of interest is considered in which a conflict can accrue between an ownership aircraft and an intruder aircraft in the presence of surrounding traffic. In this study, the heading change is considered as the action of the agent.

Previous research on conflict resolution using DRL has focused on single-agent techniques. In a single-agent setting, an agent is only concerned with the outcome of its own actions. However, the ATM system is multi-agent, and a single agent can rarely model its collective behavior. In a multi-agent domain, an agent not only receives feedback on its own actions but also on those of other agents, leading to complex learning dynamics. As agents interact concurrently, their actions constantly reshape the environment, resulting in a non-stationary environment. Consequently, learning among agents can cause changes in their policies and affect the optimal policies of others, which makes at a given point in the multi-agent setting potentially ineffective in the future. To address these challenges, researchers have focused on multi-agent DRL methods. One strategy commonly employed to tackle multi-agent environments is independent learning, where agents treat other agents as part of the environment, and there is no direct communication among them. For instance, Dong et al.[9] introduced an independent Deep Q-Network method for conflict resolution. In,[22] a DRL method for distributed conflict resolution was proposed to guarantee minimum separation of aircraft during operation. In this study, a single policy is considered and used by all agents independently. A unique policy was determined using different combinations of heading, speed, and altitude as the action space. However, this approach often encounters difficulties as each agent operates independently within the environment, leading to learning instability. Due to the concurrent learning of multiple agents, the actions taken by one agent impact the rewards received by other agents and the evolution of the environment state. In this respect, it is required to adopt a strategy that each agent is aware of decisions taken by others to take more informed actions. In the current study, we aim to address this challenge by adopting a centralized learning approach, providing a comprehensive understanding of the overall status of the environment, and enabling more efficient decisions.[16] This, in turn, facilitates the exchange of information and coordination among agents, mitigating the learning instability caused by independent learning (see[16] for a detailed explanation of the concept of centralized learning).

In this study, we present a novel cooperative framework for resolving conflicts in high-density en-route areas. The proposed approach employs a centralized learning, decentralized execution scheme, which allows for effective coordination among multiple aircraft. By incorporating information from other aircraft that are taking action in the environment during the training, we create a stationary environment for each agent, ensuring the stability of the training process. The state space for each agent includes not only its own information but also relevant details about neighboring aircraft, such as their speeds, heading angles, and closest distance from the agent's aircraft. To learn the optimal policies for agents in a continuous action space, we employ the deep deterministic policy gradient (DDPG) algorithm, which has shown promising performance in various multi-agent applications.[13,29] The ultimate goal of this study is to explore the possibility of delivering safe and climatically-friendly aerial traffic computationally efficiently. In this respect, we first optimize trajectories of fights for a real traffic scenario considering the mitigation of the climate impact induced by non-CO_2 emissions as the flight planning objective (Section 2). The feasibility of the optimized trajectories is assessed in terms of operational cost and air traffic safety. The proposed conflict resolution strategy presented in Section 3, is then implemented, and the agents are trained. Once the policies are obtained, they are adopted by aircraft in different scenarios to resolve conflicts of climate-optimal trajectories (Section 4).

2. Climate Optimal Aircraft Trajectory Planning

In order to address the problem of aircraft trajectory optimization within the context of optimal control, certain components are necessary, namely the aircraft dynamical model (or dynamical constraint), performance index (or cost functional), as well as a collection of equality and inequality boundary and path constraints.[25] In the following, we will briefly present these elements. A detailed explanation of the methodology can be found in.[2]

DOI: 10.13009/EUCASS2023-633

Let us consider the 2D point-mass model of aircraft as:[11]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \tag{1}$$

where $\mathbf{x}(t)$ is the state vector and $\mathbf{u}(t)$ is the control vector, defined as:

$$\mathbf{x}(t) = \begin{bmatrix} \varphi & \lambda & v & m \end{bmatrix}^T, \qquad \mathbf{u}(t) = \begin{bmatrix} \chi & C_T \end{bmatrix}^T.$$

In the given equation φ is the latitude, λ is the longitude, v is the true speed, and m is the aircraft mass. The control vector includes heading angle (χ) and thrust coefficient (C_T). The function $\mathbf{f}$ is a vector field that maps $\mathbf{f} : \mathbb{R}^+ \times \mathbb{R}^4 \times \mathbb{R}^2 \to \mathbb{R}^4$ is given as

$$\mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) =: \begin{bmatrix} f_1(t) & f_2(t) & f_2(t) & f_2(t) \end{bmatrix}^T$$

where

$$f_1(t) = \frac{(v\cos\chi + w_x)}{R_N + h}, \quad f_2(t) = \frac{v\sin\chi + w_y}{(R_M + h)\cos\varphi}, \quad f_3(t) = \frac{1}{m}(T(C_T) - D(m, v)), \quad f_4(t) = -f_c(C_T). \tag{2}$$

In Eq. (2), h represents the altitude which is a constant value indicating that the optimization will be performed only on the lateral path and the altitude profile will be unchanged during flight. R_M and R_N refer to the radii of curvature of the ellipsoid and the meridian and prime vertical, respectively. f_c represents the fuel burn rate, T and D are the magnitudes of thrust and drag forces, and finally, (w_x, w_y) corresponds to the components of the wind.

In order to account for both physical and operational factors, we establish the following set of constraints on thrust coefficient, the Mach number (M), and calibrated airspeed (v_{CAS}):

$$\begin{aligned} v_{CAS,stall} \leq & v_{CAS}(v) \leq v_{CAS,\max} \\ C_{T,\min} \leq & C_T \leq C_{T,\max} \\ M(v) \leq & M_{\max} \end{aligned} \tag{3}$$

Additionally, the following boundary conditions are imposed on the initial and final values of aircraft's states:

$$\begin{aligned} t(0) &= t_0 \\ [\varphi, \lambda, v, m](0) &= [\varphi_0, \lambda_0, v_0, m_0] \\ [\varphi, \lambda, v](t_f) &= [\varphi_f, \lambda_f, v_f]. \end{aligned} \tag{4}$$

It should be noted that the final mass of aircraft is not specified and will be optimized within optimization.

In order to determine climate-optimized trajectories, it is necessary to include the climate impact as an objective in the cost functional of the optimal control problem. To quantify the climate impact, we utilize the algorithmic climate change functions. These functions, developed within the EU-projects ATM4E and FlyATM4E, require specific meteorological variables as input and provide a quantification of the climate impact in terms of the average temperature response (ATR) over a 20-year period. For a detailed explanation of these functions, interested readers are referred to.[8] Making use of aCCFs to model climate effects, we define the objective function for the flight planning problem as follows:

$$\begin{aligned} J &= C_{SOC} + \mathrm{EI} \cdot 10^{10} \cdot C_{ATR} \\ C_{SOC} &= 0.51 \cdot [t_f - t_0] + 0.75 \cdot [m(t_0) - m(t_f)] \\ C_{ATR} &= \int_{t_0}^{t_f} \sum_{j=1}^{5} \mathrm{ATR}_j^{\mathrm{mean}}(t, \mathbf{x}(t), \mathbf{u}(t))\mathrm{d}t. \end{aligned} \tag{5}$$

Here, C_{SOC} and C_{ATR} represent the simple operating cost (SOC) (in USD)[28] and average temperature response (in Kelvin), respectively.[2] $\mathrm{ATR}_j^{\mathrm{mean}}$ denotes the computed ATR for species j (where $j \in \{\mathrm{CO_2}, \mathrm{Cont.}, \mathrm{CH_4}, \mathrm{O_3}, \mathrm{H_2O}\}$) by considering the mean values of an ensemble weather forecast for the required meteorological variables (e.g., temperature and relative humidity). As indicated in Equation (5), the considered cost function incorporates both climate impact and operational cost. Generally, there exists a trade-off between these two objectives, and the weighting parameter called the environmental index (EI [USD/K]) serve to prioritize ATR over the operating cost.

So far, we have defined all the elements required to formulate aircraft trajectory optimization problem in the framework of optimal control theory. To solve the defined optimization problem, direct optimal control, as a suitable approach to deal with problems with high nonlinearity in the dynamical model, cost function and constraint, is employed.[5]

DOI: 10.13009/EUCASS2023-633

All in all, the optimized trajectory for the flight i-th (i.e., trajectory-level optimization) is obtained as:

$$T_i^o := (\varphi_i^o, \lambda_i^o, v_i^o).$$

Once the climate-optimal trajectories for individual aircraft are determined, the next step is to integrate them into the network-scale traffic and their performance is then evaluated based on the number of conflicts. In the next section, we present the proposed framework to compensate for the arisen conflicts of adopting climate-optimized trajectories.

3. Conflict Resolution using Deep Deterministic Policy Gradient Algorithm

In the context of aviation, a conflict situation refers to an event where two aircraft come closer to each other than a predetermined minimum distance, either horizontally or vertically. To ensure safety, an aircraft operating in airspace establishes a forbidden zone around itself in the shape of a cylinder. The dimensions of this cylinder are determined by the required minimum horizontal and vertical separation between two aircraft. If an intruder aircraft enters the restricted area of the aircraft, a potential conflict situation is detected. Subsequently, modifications to the aircraft profiles become necessary to resolve the detected conflict. In this study, this sequential decision-making process, independent of prior history, is modeled as a Markov Decision Process (MDP) and solved using reinforcement learning techniques.

3.1 Reinforcement Learning

Reinforcement Learning (RL) models the try-and-error learning process as an MDP. In RL, the agent serves as the central component of the system, operating within an environment. In this framework, any decision-maker is considered an agent, while everything outside is considered part of the environment. The interaction between the agent and environment is defined by the tuple (S, A, P, R, λ).[20] At the time t, the agent receives the state of the environment $s_t \in S$, and selects an action $a_t \in A$ based on the received state. The environment responds to this action by transitioning to the next state $s'_t \in S$ according to the state transition function P and providing the agent a feedback reward $r(s_t, a_t)$. P is the state transition function defined as $P(s'_t|s_t, a_t) : S \times A \times S \rightarrow [0, 1]$, which represents the probability of transitioning from the current state $s_t \in S$ to the next state $s'_t \in S$, given that action $a_t \in A$ is taken. This approach, similar to human learning, guides the agent towards enhancing its future choices to maximize its future rewards. The decisions made by the agent are determined by a policy π, which maps the states of the environment to the corresponding actions to be taken. The main objective of the agent is to find an optimal policy π^* that maximizes the expected future rewards:[20]

$$R = \mathbb{E}[\sum_{t=0}^{T} \lambda^t r(s_t, a_t)], \tag{6}$$

where T is the time horizon. The parameter $\lambda \in [0, 1]$ is a discount factor that specifies the importance of short-term or long-term rewards.

3.2 Conflict Resolution Modeling

In this study, the conflict resolution problem is formulated within the framework of RL, where each aircraft in the airspace is considered an agent. Within this system, each aircraft must make decisions regarding its flight profile to avoid interactions with other aircraft. In the following, we describe the state space, action space, and reward function for the agents in detail.

State space

The state of each aircraft is defined based on the required information to make decisions. For each aircraft operating in the airspace at time t, the state contains the current speed of the aircraft and the heading angle. These parameters capture the essential flight information necessary for decision-making. Since this is a multi-agent environment involving multiple decision-makers, it is important to consider communication between aircraft to inform aircraft about the surrounding traffic. To facilitate this, each aircraft is provided with local information about the traffic, which serves as its state representation. In this respect, m closest aircraft are considered as the neighboring aircraft, and their speed profile, heading angle, and the minimum distance from the ownership aircraft until the next time step $t + 1$ are included in the state of the ownership. From this follows, at the time slot t, for the aircraft k, the state is described as:

$$s_t^k = \{\chi_t^k, v_t^k, I_t^1, ..., I_t^m\} \tag{7}$$

DOI: 10.13009/EUCASS2023-633

where χ_t^k and v_t^k are the aircraft's heading and speed at time t, and $I_t^m = (\chi_t^m, v_t^m, \mathbf{los}_t(m,k))$ is the information of aircraft m, and $\mathbf{los}_t(m,k)$ is the minimum distance of aircraft m to the aircraft k. By incorporating the loss of separation distance between aircraft as part of the state space, the agents can develop strategies to maintain a safe distance from neighboring aircraft. The loss of separation distance serves as a critical metric to assess the level of proximity between aircraft and is a key component of the state representation.[3]

Action space

Main maneuvers for resolving conflicts include speed change, heading angle change, departure time change (or cancellation), and altitude change. A fully automated conflict resolution system can use all these maneuvers to perform the resolution efficiently. Within the framework of this study, we consider the speed (Mach) profile as the only decision variable. Then, the action space is defined as:

$$A = \{\zeta | \zeta \in [-0.06, 0.04]\} \tag{8}$$

The action space is continuous, allowing the agent to select any value between -0.06 and 0.04 to adjust its current Mach number.

Reward function

As a result of changing the speed of aircraft, the environment transits to a new state, and each aircraft experiences interactions with other aircraft. The reward function in the proposed method is designed to balance the objectives of safety assurance and minimum deviations from the optimized trajectory (i.e., at the trajectory level). The reward with respect to safety is expressed as

$$CC_t^k = \sum_{q=1,q\neq k}^{N} \sum_{\tau=t}^{t+1} -c_\tau^{kq} \qquad c^{kq} = \begin{cases} 1 & \text{if} \quad d^{kq} < D_0 \quad \text{and} \quad h^{kq} < H_0 \\ 0 & \text{else} \end{cases} \tag{9}$$

where d_τ^{kq} and h_τ^{kq} are the horizontal and vertical distances between aircraft k and q at time $\tau \in [t, t+1]$, respectively. Considering the fact that deviating from the cruise speed degrades the optimal performance achieved at the trajectory level, it is important to penalize speed adjustments while aiming at resolving conflicts. In this respect, a negative term in the reward function is defined to minimize the speed changes:[30]

$$CV_t^k = -\sum_{\tau=t}^{t+1} 0.001 \times e^{-\omega_v^2}; \qquad \omega_v := (v_c - v_o)/(v_{max} - v_{min}) \tag{10}$$

where v_c is the modified true speed, v_o is the original true speed, and v_{min} and v_{max} are the minimum and maximum allowable true speed change, respectively. The total reward for aircraft k at time t is expressed as:

$$r_t^k = CC_t^k + CV_t^k \tag{11}$$

which encourages the agents to prioritize actions that minimize potential conflicts and maintain appropriate separation distances while still considering the need to adhere to the optimized trajectories. The goal of each aircraft is to approach the destination while maximizing its own expected rewards $R^k = \sum_{t=0}^{T_f} (\lambda)^t r_t^k$.

3.2.1 Multi Agent Deep Deterministic Policy Gradient Algorithm

Reinforcement learning algorithms have been successfully used in solving MDP problems with promising results.[20] However, applying the single-agent RL methods to the multi-agent setting by considering each agent to learn independently may result in poor performance.[17] Since all agents interact concurrently and update their policies independently, their actions constantly reshape the environment, and we will face the problem of a non-stationary environment. This causes instability in learning as it violates the stationary assumption required for the convergence of single-agent reinforcement learning algorithms.[4] In this respect, we propose a multi-agent framework to resolve the conflicts between aircraft. To overcome the problem of the non-stationary environment, we rely on a centralized training and decentralized execution scheme where aircraft share their information during the training process. However, this information is not used for the test. The RL algorithm to solve this problem is the Deep Deterministic Policy Gradient (DDPG), an off-policy method that performs well across various challenging environments.[20]

The Deep Deterministic Policy Gradient is a widely recognized actor-critic algorithm that combines Deterministic Policy Gradient (DPG) with Deep Q-Network (DQN) to solve complex problems with continuous action space.[20]

DOI: 10.13009/EUCASS2023-633

The DDPG algorithm comprises an actor network and a critic network. The critic uses a parameterized action-value function (Q^{η}_{θ}) to estimate the expected rewards associated with the deterministic policy $\eta_{\vartheta} : S \rightarrow A$. The actor will adjust the parameters ϑ of the policy η_{ϑ} in accordance with the direction that the critic suggests. Both parameterized critic and actor are utilized in the training phase, however, for the execution, only the actor will be used.[20] The DDPG stores the experiences in a reply buffer (Ξ) and then uses the sample trajectories from the buffer for the training process to improve data efficiency. To stabilize the training, two target networks $\hat{Q}^{\hat{\eta}}_{\hat{\theta}}$ and $\hat{\eta}_{\hat{\vartheta}}$ identical to the original networks, called target actor and target critic, respectively, are created, and their parameters are periodically updated by copying θ and ϑ from the original networks. The aim is to find the optimal policy parameterized by ϑ which maximize the objective $J(\vartheta) = \mathbb{E}_{s\sim\Xi}[Q^{\eta}(s,a)|_{a=\eta_{\vartheta}(s)}]$. Accordingly, the gradient of objective J can be written as follows:

$$\nabla_{\vartheta} J(\vartheta) = \mathbb{E}_{s\sim\Xi}[\nabla_{\vartheta}\eta_{\vartheta}(a|s)\nabla_{a}Q^{\eta}(s,a)|_{a=\eta_{\vartheta}(s)}] \tag{12}$$

Since within this formulation, $\nabla_{a}Q^{\eta}(s,a)$ is required, the policy η needs to be continues. Specifically, the critic network is updated based on:

$$L(\theta) = \mathbb{E}_{(s,a,s',r)\sim\Xi}[(Q^{\eta}_{\theta}(s,a) - (r + \lambda\hat{Q}^{\hat{\eta}}_{\hat{\theta}}(s',a')))^2] \tag{13}$$

where $a' = \hat{\eta}_{\hat{\vartheta}}(s')$.

Within the framework of multi-agent DDPG, each agent is modeled as a DDPG agent, which has access to the state and actions of the other agents during the training. In particular, the actor network, which maps the local state of the agent to the optimal action, gets as input only the state of the agent to provide the actions. However, the critic network uses the states and actions of all agents to evaluate the actions generated by the actor network.[16] To go further detail to the Multi agent DDPG algorithm, let consider the actor η^k and centralized critic network Q^k parameterized by ϑ^k and θ^k for agent k. At each time step t the for each agent, the initial state s^k_t is fed to the associated actor network to generate action $a^k_t = \eta^k(s^k_t)$. The states of all agents together with generated actions are applied to the environment, resulting in transition of agents to new states and obtaining rewards. The transition is stored in a reply buffer as tuple (O, Λ, O', R), where $O : \{s^1, s^2, ..., s^N\}$, $\Lambda : \{a^1, a^2, ..., a^N\}$, $O' : \{s'^1, s'^2, ..., s'^N\}$, $R : \{R^1, R^2, ..., R^N\}$ are the states, actions, next states and obtained rewards of all agents, respectively. Since each Q^k is learned separately, each agent can have arbitrary reward structures. Once one episode is done (i.e., all agents approach their final states), for each agent, the η^k and Q^k are trained using N_{batch} samples of replay buffer Ξ.

Suppose that Ξ_m be mini-batch from Ξ. Each agent updated the parameters of its centralized critic network by minimizing the following loss

$$L(\theta^k) = \mathbb{E}_{(O,\Lambda,O',R)\sim\Xi_m}[(Q^k(O,\Lambda) - y^k)^2] \tag{14}$$

where $Q^k(O,\Lambda)$ is the predicted value of the critic network and y^k is target value obtained as

$$y^k = r^k + \gamma Q'^k(O', a'^1, a'^2, ..., a'^N)|_{a'^k=\eta'^k(s')} \tag{15}$$

where a'^k and $Q'^k(O', a'^1, a'^2, ..., a'^N)$ are the predicted action and value by target actor and target critic, respectively. Once the parameters of critic network is updated, the parameters of actor network are updated by maximizing the gradient of the expected return as

$$\nabla_{\vartheta^k} J(\eta_{\vartheta^k}) = \mathbb{E}_{O\sim\Xi_m}[\nabla_{\vartheta^k}\eta_{\vartheta^k}(s^k)\nabla_{a^k}Q^{\eta^k}(O,\bar{\Lambda}) \tag{16}$$

where $\bar{\Lambda} : \{\eta^1(s^1), \eta^2(s^2), ..., \eta^N(s^N)\}$. After training the parameters of target networks are updated as follow:

$$\hat{\vartheta}^k \leftarrow \alpha\vartheta^k + (1-\alpha)\hat{\vartheta}^k \tag{17}$$

$$\hat{\theta}^k \leftarrow \alpha\theta^k + (1-\alpha)\hat{\theta}^k \tag{18}$$

where α is the learning rate. Once the training process is done, the trained critic model is removed. Then at each time step t, only the local state of the agent k is required to be inputted to the trained actor model η^{*k} to obtain the optimal action $a^k_t = \eta^{*k}(s^k_t)$.

4. Results

This paper presents a case study involving real-time traffic in Spanish airspace over a one-hour period. The study focuses on a specific traffic scenario on June 18, 2018, between 12:00 and 13:00. The flight data, including the entry time of aircraft into the designated airspace (latitude: [35,44], longitude: [-12,4]) and the most frequently used flight level during the cruise phase, was extracted from the DDR2 dataset Meteorological data necessary for the optimization

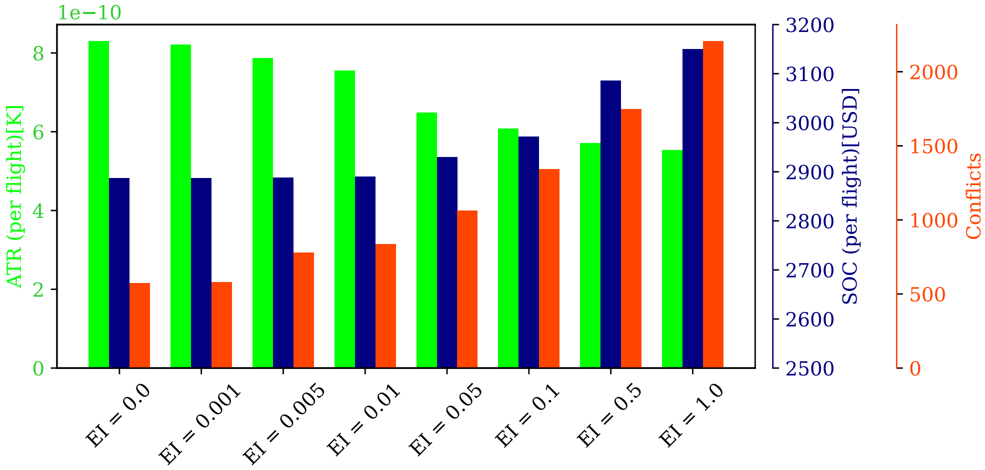

Figure 1: Results for different sets of trajectories in terms of ATR, SOc, and number of conflicts. For ATR and SOC, the results presented per flight.

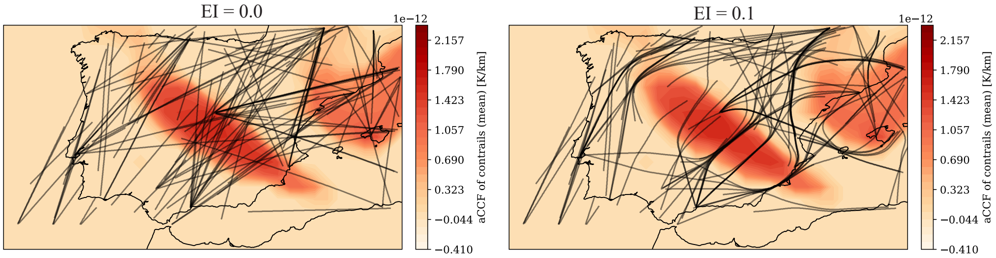

Figure 2: Lateral routes of aircraft for two different EI values at FL380.

process were collected from the ERA5 reanalysis data product. All flights that crossed the airspace within the specified time window were optimized using the method described in Section 2 for various routing options (through changing the parameter penalizing the climate effects, i.e., EI). The Trapezoidal rule is implemented to transcribe the formulated optimal control problem to nonlinear programming, which is then solved using the interior point method (using IPOPT in Python).

Figure (1) represents the climate impact, operating cost, and the number of conflicts for various sets of climate-optimal trajectories. The climate impact of different trajectory sets is evaluated in terms of ATR. From Fig. (1), it can be seen that trajectory planning can efficiently mitigate the climate impact. For instance, in the case of EI = 1.0, the climate impact can be reduced by 33%. While trajectory planning offers great potential to reduce the climate impact of aviation, it is essential to evaluate other performance indicators such as operational cost and air traffic safety. For the cost-efficiency assessment, we adopt SOC which is a weighted sum of flight time and fuel consumption. Analyzing the results shows that when trajectories with lower climate impact are adopted, the operating cost and number of conflicts increase. This is because of the tendency of aircraft to avoid climate hotspots, which might yield longer routes. Therefore, such hotspot avoidance behavior, while beneficial from a climate impact perspective, increases the operational cost.

In addition to the operating cost, the traffic concentration around climate hotspots leads to increased congestion. The congestion arises as multiple aircraft navigate around the hotspots, resulting in proximity and potential conflicts between flights. In Fig. (2), the optimized lateral paths associated with cost-optimal and climate-optimal (considering EI = 0.1) routing strategies are depicted with the aCCF of contrails as the colormap (due to its dominant climate effects[8]). By comparing two sets of trajectories at the same flight level, it can be seen that trajectories with less climate impact tend to avoid contrail-sensitive areas, resulting in congestion. By adopting trajectories with higher climate impact mitigation potential, e.g., EI = 1.0, a critical increase (by the factor of four compared to cost optimal scenario) in the number of conflicts is observed (see Fig. (1)).

As demonstrated, the mitigation of climate impact comes at the cost of a substantial rise in conflict occurrences for higher EIs. Resolving these conflicts at the tactical level is challenging, resulting in the infeasibility of the optimized trajectories. Therefore, to deliver operationally feasible climate-optimal trajectories, traffic complexity needs to be

DOI: 10.13009/EUCASS2023-633

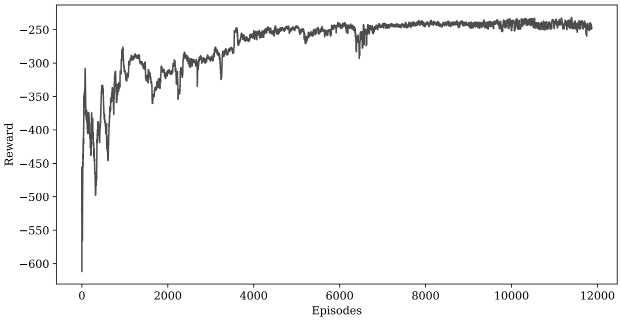

Figure 3: Learning curve of the proposed approach on the Case Study.

Table 1: The set of training parameters and their values

Parameter	Value
Time step (t)	900 s
Discount factor for future rewards (γ)	0.98
Learning rate of updating target networks for both models (α)	10^{-2}
Size of mini-batch (l)	20
Exploration noise magnitude	0.5
maximum time step in each episode (t_{max})	4
Learning rate of actor model (lr_a)	10^{-4}
Learning rate of critic model (lr_c)	10^{-3}

reduced at the strategic level. For this aim, we utilize the resolution method described in Section 3 to strategically increase the manageability of the traffic, focusing on reducing the encountered conflicts.

To analyze the performance of the proposed framework, one set of trajectories is selected for the training and the rest for test cases. For the train set, as explained in Section 3, one actor and one critic network are assigned to each agent. To improve the stability of the training, two target networks identical to the original actor and critic are created. The target networks' weighting parameters were copied from the original networks at each training step. Each critic network is considered as a fully connected neural network consisting of two hidden layers with 300 nodes in the first layer and 100 nodes in the second hidden layer. The actor networks also are two fully connected networks with 100 nodes in the first and second layers. For the hidden layers, the ReLU activation function was applied. The actor's output utilized the tangent activation function, while the critic's output employed the linear activation function. The Adam optimizer was utilized for both the actor and critic loss.

The state space, action space, and reward function described in Section 3 have been implemented. The training process is performed based on the aircraft trajectories associated with EI = 0.01. This particular selection was made because this set represents a challenging scenario with a high number of conflicts, albeit not as complex as EI = 1.0. Each episode in training represents one hour of traffic. A time step of t = 15 min is considered, which means that every 15 min, the aircraft receives an observation from the environment and can take action accordingly. While observations are collected every 15 minutes, they encompass the flight intentions for the subsequent 15 minutes. All flights within the considered time frame (12:00-13:00) are considered in the training process. However, only aircraft involved in conflicts are permitted to modify their speed profiles. Flight profiles of the aircraft without conflicts remain unchanged, while conflict resolution becomes the responsibility of the involved aircraft. During the testing phase, the conflicting aircraft adopt the trained policies. It is worth noting that assigning a policy to each individual aircraft, whether in conflict or not, is also a possible option. However, this would introduce additional complexity to the problem. Moreover, from an operational perspective, aircraft not engaged in conflicts must adhere to their optimal trajectories.

The learning curve, depicted in Fig. (3), illustrates the episode returns obtained during the training stage. Figure (3) provides insight into the convergence of the proposed framework toward the optimal policy. It can be seen that the method is able to converge toward actions that resolve a high number of conflicts.

Once the policies are obtained, they are adopted by the conflicting aircraft in different test scenarios. Seven sets of the trajectory (i.e., EI = 0.0, 0.001, 0.005, 0.05, 0.1, 0.5, 1.0) are selected for test cases. Figure (4) presents the results for different test scenarios. It can be seen that the learned policies can efficiently be used to resolve more than 40% conflicts in most cases. It is worth mentioning that, as in this study, we only use speed advisory as the action apace; the head-to-head conflicts cannot be resolved simply by changing speed, but changing altitude or/and rerouting

DOI: 10.13009/EUCASS2023-633

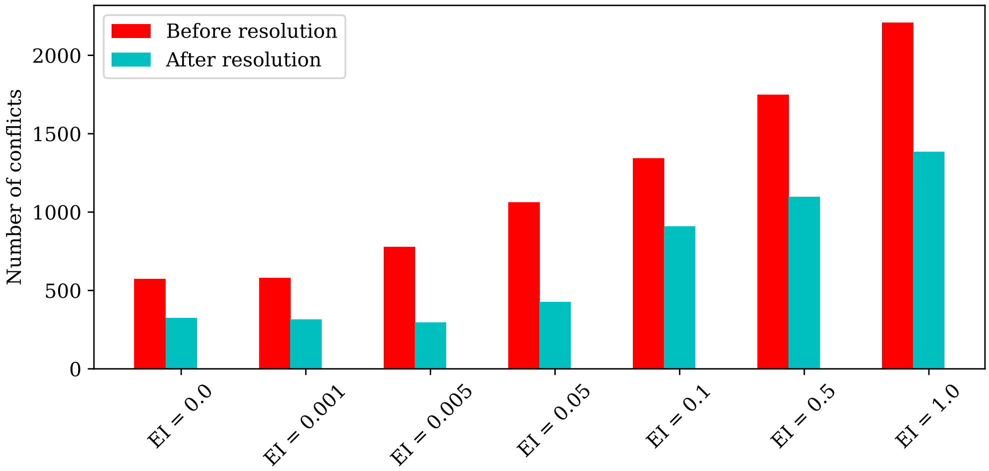

Figure 4: Test case studies using the trained models.

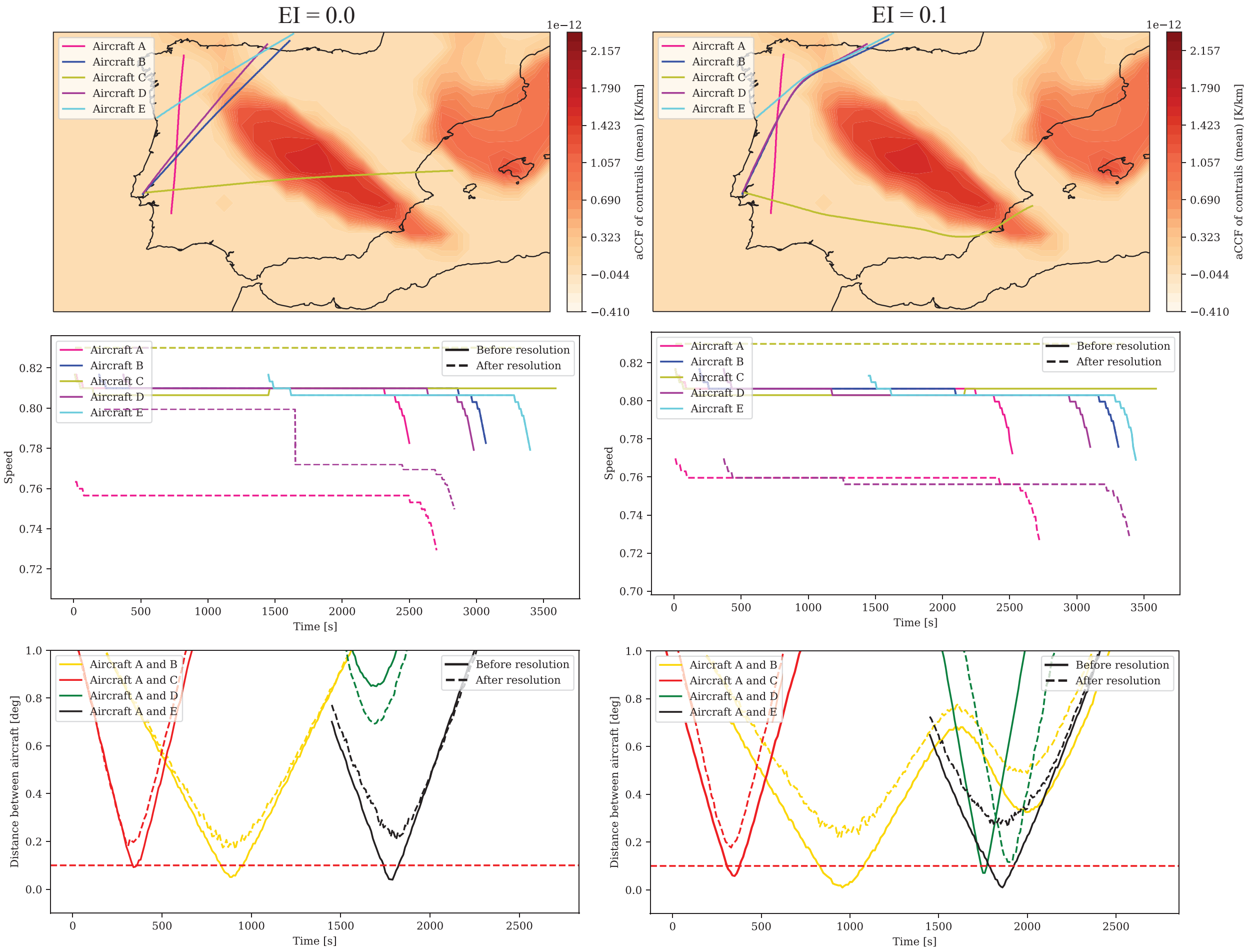

Figure 5: Lateral paths of a set of conflicting aircraft, and speed profiles, and the relative distance before and after applying the trained policies.

DOI: 10.13009/EUCASS2023-633

are required for those conflicting flights.

A specific set of conflicting aircraft is chosen from two different sets to analyze their profiles and evaluate the proposed algorithm's performance. This set comprises Aircraft A, which serves as the reference aircraft, and all other aircraft that are in conflict with it (i.e., B, C, D, E). Although Aircraft B, C, D, and E may also be involved in conflicts with other aircraft, this particular example focuses solely on Aircraft A. Figure (5) illustrates the lateral route, speed profile, and the distances between Aircraft A and the other aircraft. The route of Aircraft A intersects with four other aircraft, potentially leading to conflicts. It can be observed that at certain times, the distances between the aircraft are less than $0.1°$, indicating the occurrence of conflicts. Each conflicting aircraft is assigned a policy to resolve the conflicts. By executing the assigned policies, the speeds of the aircraft are adjusted to resolve the conflicts. The speed profiles of the aircraft before and after the conflict resolution are depicted in Figure (5). It is evident that the implemented policies effectively resolve a high number of conflicts.

The study has demonstrated that trained policies are highly effective in resolving conflicts, even in complex scenarios, with acceptable performance. Conflict resolution with conventional algorithms is time-consuming and impractical for real-time applications, as they require to be performed from scratch, and their execution time increases with the number of aircraft involved and airspace congestion. The key advantage of using trained policies is their versatility, as they can be applied to any set of trajectories with promising results. It is important to note that this study focused on one decision variable to resolve the conflicts. However, it is expected that more conflicts will be resolved by incorporating additional decision variables (e.g., lateral path, altitude) due to the provided flexibility. This direction will be explored in future studies.

5. Conclusion

In this study, a conflict resolution method was presented to be used in distributed air traffic systems for complex traffic patterns with uneven distribution. The approach utilized a multi-agent deep reinforcement learning algorithm in which each aircraft was treated as a decision-maker. The proposed method employed the Deep Deterministic Policy Gradient algorithm to find optimal policies in a centralized learning, decentralized execution framework. In this respect, each agent was informed of the decisions made by other agents during the training to create a more stable learning environment. To evaluate the proposed approach, for a real traffic scenario, the trajectories were optimized in a climate-friendly manner considering different priority levels for climate impact mitigation. The obtained policies were applied to aircraft in different sets to map the observations of each agent to optimal action. The results demonstrated that even in situations where climate-optimal trajectories caused congestion in specific areas, the proposed algorithm could efficiently resolve many conflicts. The presented study revealed that the proposed framework has the potential to facilitate the delivery of climate-optimal trajectories by balancing environmental concerns and air traffic safety.

Future works

This work should be extended to develop a more comprehensive algorithm by incorporating additional decision variables, such as departure time, lateral path, and altitude profile optimization, to further enhance conflict resolution efficiency and provide additional degrees of freedom. Furthermore, it is important to conduct the same analysis for 4D climate optimal trajectories, where the altitude profile is also optimized in addition to the lateral route, within the current structured airspace. Additionally, other deep reinforcement learning algorithms should be considered and compared to evaluate their performance. Comparing the effectiveness of different algorithms will provide valuable insights. Moreover, exploring the potential of transfer learning by training a policy with one set of trajectories and then retraining the model using data from multiple days can be explored as a means to develop a robust model applicable to a variety of unseen scenarios.

6. Acknowledgments

This research was carried out as a part of the EU-Project RefMAP. RefMAP has received funding from the Horizon Europe program 2023 under grant agreement No 101096698. The opinions expressed herein reflect the authors' views only. Under no circumstances shall the Horizon Europe program be responsible for any use that may be made of the information contained herein.

DOI: 10.13009/EUCASS2023-633

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[2] Fateme Baneshi, Manuel Soler, and Abolfazl Simorgh. Conflict assessment and resolution of climate-optimal aircraft trajectories at network scale. *Transportation Research Part D: Transport and Environment*, 115:103592, 2023.

[3] Marc Brittain, Xuxi Yang, and Peng Wei. A deep multi-agent reinforcement learning approach to autonomous separation assurance. *arXiv preprint arXiv:2003.08353*, 2020.

[4] Lorenzo Canese, Gian Carlo Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spanò. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11):4948, 2021.

[5] Benoit Chachuat. Nonlinear and dynamic optimization: From theory to practice. Technical report, 2007.

[6] Rohan Chopra and Sanjiban Sekhar Roy. End-to-end reinforcement learning for self-driving car. In *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018, Volume 1*, pages 53–61. Springer, 2020.

[7] Valentin Courchelle, Manuel Soler, Daniel González-Arribas, and Daniel Delahaye. A simulated annealing approach to 3D strategic aircraft deconfliction based on en-route speed changes under wind and temperature uncertainties. *Transportation research part C: emerging technologies*, 103:194–210, 2019.

[8] Simone Dietmüller, Sigrun Matthes, Katri Dahlmann, Hiroshi Yamashita, Manuel Soler, Abolfazl Simorgh, Florian Linke, Benjamin Lührs, Maximiliam M. Meuser, Christian Weder, Feijia Yin, Federica Castino, and Volker Grewe. A python library for computing individual and merged non-co$_2$ algorithmic climate change functions: CLIMaCCF v1.0. *Geoscientific Model Development (under review)*, 2022.

[9] SUI Dong, XU Weiping, and Kai Zhang. Study on the resolution of multi-aircraft flight conflicts based on an idqn. *Chinese Journal of Aeronautics*, 35(2):195–213, 2022.

[10] Kaitano Dube, Godwell Nhamo, and David Chikodzi. Covid-19 pandemic and prospects for recovery of the global aviation industry. *Journal of Air Transport Management*, 92:102022, 2021.

[11] Daniel González-Arribas, Manuel Soler, Manuel Sanjurjo-Rivo, Javier García-Heras, Daniel Sacher, Ulrike Gelhardt, Juergen Lang, Thomas Hauf, and Juan Simarro. Robust optimal trajectory planning under uncertain winds and convective risk. In *ENRI International Workshop on ATM/CNS*, pages 82–103. Springer, 2017.

[12] Eulalia Hernández Romero. Probabilistic aircraft conflict detection and resolution under the effects of weather uncertainty. 2020.

[13] Liwei Huang, Mingsheng Fu, Hong Qu, Siying Wang, and Shangqian Hu. A deep reinforcement learning-based method applied for solving multi-agent defense and attack problems. *Expert Systems with Applications*, 176:114896, 2021.

[14] David S Lee, DW Fahey, Agniezka Skowron, MR Allen, Ulrike Burkhardt, Q Chen, SJ Doherty, S Freeman, PM Forster, J Fuglestvedt, et al. The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018. *Atmospheric Environment*, 244:117834, 2021.

[15] Sheng Li, Maxim Egorov, and Mykel Kochenderfer. Optimizing collision avoidance in dense airspace using deep reinforcement learning. *arXiv preprint arXiv:1912.10146*, 2019.

[16] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[17] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE transactions on cybernetics*, 50(9):3826–3839, 2020.

[18] Mercedes Pelegrín and Claudia dâAmbrosio. Aircraft deconfliction via mathematical programming: Review and insights. *Transportation science*, 56(1):118–140, 2022.

DOI: 10.13009/EUCASS2023-633

[19] Duc-Thinh Pham, Phu N Tran, Sameer Alam, Vu Duong, and Daniel Delahaye. Deep reinforcement learning based path stretch vector resolution in dense traffic with uncertainties. *Transportation research part C: emerging technologies*, 135:103463, 2022.

[20] Sudharsan Ravichandiran. *Deep Reinforcement Learning with Python: Master classic RL, deep RL, distributional RL, inverse RL, and more with OpenAI Gym and TensorFlow*. Packt Publishing Ltd, 2020.

[21] Marta Ribeiro, Joost Ellerbroek, and Jacco Hoekstra. Improvement of conflict detection and resolution at high densities through reinforcement learning. *Proceedings of the ICRAT*, 2020.

[22] Marta Ribeiro, Joost Ellerbroek, and Jacco Hoekstra. Distributed conflict resolution at high traffic densities with reinforcement learning. *Aerospace*, 9(9):472, 2022.

[23] MJ Ribeiro. Conflict resolution at high traffic densities with reinforcement learning. 2023.

[24] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[25] Abolfazl Simorgh, Manuel Soler, Daniel González-Arribas, Sigrun Matthes, Volker Grewe, Simone Dietmüller, Sabine Baumann, Hiroshi Yamashita, Feijia Yin, Federica Castino, et al. A comprehensive survey on climate optimal aircraft trajectory planning. *Aerospace*, 9(3):146, 2022.

[26] Abolfazl Simorgh, Manuel Soler, Daniel GonzÃ¡lez-Arribas, Sigrun Matthes, Volker Grewe, Simone DietmÃ¼ller, Sabine Baumann, Hiroshi Yamashita, Feijia Yin, Federica Castino, Florian Linke, Benjamin LÃ¼hrs, and Maximilian Mendiguchia Meuser. A comprehensive survey on climate optimal aircraft trajectory planning. *Aerospace*, 9(3), 2022.

[27] Zhuang Wang, Weijun Pan, Hui Li, Xuan Wang, and Qinghai Zuo. Review of deep reinforcement learning approaches for conflict resolution in air traffic control. *Aerospace*, 9(6):294, 2022.

[28] Hiroshi Yamashita, Feijia Yin, Volker Grewe, Patrick Jöckel, Sigrun Matthes, Bastian Kern, Katrin Dahlmann, and Christine Frömming. Newly developed aircraft routing options for air traffic simulation in the chemistry–climate model EMAC 2.53: AirTraf 2.0. *Geoscientific Model Development*, 13(10):4869–4890, 2020.

[29] Ziming Yan and Yan Xu. A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system. *IEEE Transactions on Power Systems*, 35(6):4599–4608, 2020.

[30] Peng Zhao and Yongming Liu. Physics informed deep reinforcement learning for aircraft conflict resolution. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8288–8301, 2021.