Direct-to Initial Approach Fix: A Reinforcement Learning Approach for Conflict-Free Arrival Sequencing in a Multi-Airport System

Hao Jiang, Weili Zeng
College of Civil Aviation
Nanjing University of Aeronautics and Astronautics
Nanjing, China
{jianghao_cca, zwlnuaa}@nuaa.edu.cn

Zhi Jun Lim, Duc-Thinh Pham, Imen Dhief, Sameer Alam
Air Traffic Management Research Institute
Nanyang Technological University
Singapore, Singapore
{zhijun.lim, dtpham, imen.dhief, sameeralam}@ntu.edu.sg

Haozhe Wang
Approach Control Unit
Tianjin Air Traffic Management Sub-Bureau
Tianjin, China
howard2400@outlook.com

Wenbin Wei
College of Engineering
San José State University
California, United States of America
wenbin.wei@sjsu.edu

***Abstract*—Terminal Manoeuvring Area (TMA) is a key air traffic subsystem that bridges en-route airspace and airport control zone. One of the main tasks of Air Traffic Control Officers (ATCOs) responsible for the TMA is to ensure that consecutive landing aircraft have the required horizontal separation. To achieve this goal, ATCOs need to make real-time decisions regarding the sequencing and spacing of arrival aircraft during daily operations, which is a primary source of their workload. Relying solely on ATCOs to make these decisions has led to issues such as delayed decision-making, excessive flight distances, and frequent trajectory adjustments, particularly in the more complex environment of multi-airport systems. To support ATCOs in making real-time decisions regarding the safe sequencing of arrival flights, this paper proposes a Reinforcement Learning approach to suggest arrival direct-to routes while considering the convergence of arrival flights destined for the same airport and conflicts with arrival flights destined for adjacent airports. A method for accelerating reinforcement learning training is also explored. Experimentation on Tianjin TMA in China shows that the proposed approach achieves conflict-free operations without sacrificing operational efficiency, and reduces training time of the RL model by 82% without compromising model performance. The results of this work demonstrates the potentials of Artificial Intelligence (AI) systems as decision-support tools in the field of Air Traffic Management (ATM).**

***Keywords*—Air traffic management, Terminal maneuvering area, Multi-airport system, Arrival sequencing, Conflict resolution, Direct-to decision making, Reinforcement learning**

I. Introduction

Terminal Manoeuvring Area (TMA) is a control area normally established in the vicinity of one or more major airports [1]. In the TMA, arrival flights need to descend in altitude, and flights coming from different directions must eventually merge into one or more landing queues. Each pair of aircraft in these queues must meet the specified horizontal separation requirements. Standard Terminal Arrival Route (STAR) is a designated Instrument Flight Rule (IFR) arrival route [1] by which arrival flight should proceed from the en-route phase to an Initial Approach Fix (IAF). So, the early research assumed that arrival flights in TMA strictly follow STARs [2]. Since there is no vertical separation between consecutive flights landing on the runway, the following flight must maintain a certain horizontal distance from the preceding flight to meet wake turbulence separation requirements. In addition, a landing aircraft will not normally be permitted to cross the runway threshold on its final approach until all preceding landing aircraft are clear of the runway-in-use [1]. Therefore, at the convergence point on the final approach leg before landing, the spacing between aircraft pairs must meet the required horizontal separation for consecutive landings. Assuming all flights strictly adhere to the STARs, it cannot be ensured that each flight will meet the required horizontal separation upon reaching the convergence point, as their estimated arrival times at the convergence point have not been intervened.

The analysis of historical arrival trajectories within the TMA indicates that arrival flights do not strictly adhere to STARs. The authors conducted a clustering study of arrival trajectories within the TMA at Nanjing Lukou Airport (ZSNJ), in China [3]. The number of clusters obtained for arrival trajectories exceeded the number of STARs, and the trajectories within each cluster also exhibited certain deviations from the corresponding STAR. Similarly, at Congonhas Airport in Brazil, there are trajectories within the same cluster that deviate significantly from the cluster centroid as well [4]. Aside from factors where weather conditions render STARs unavailable, Air Traffic Control (ATC) interventions for arrival flight sequencing, merging and spacing within the TMA often

SESAR Innovation Days 2024
12 - 15 November 2024. Rome

cause trajectory deviations from STARs [4]. For multi-airport systems, the significant inter-airport operational dependency further increases traffic complexity [5], leading to more diverse arrival trajectory patterns.

The emergence of trajectories different from STARs is entirely the result of Air Traffic Control Officers (ATCOs) interventions during the tactical phase. The ATCOs have two primary intentions for intervening in horizontal trajectories of arrival flights: 1) applying route stretching to delay flights; and 2) applying route shortening to facilitate early landings. Vectoring is a flexible tactical strategy that can be used for both route stretching and shortening. When executing vectoring, ATCOs need to provide a series of heading instructions, e.g. Fly heading 180, until the arrival flight intercepts the Instrument Landing System (ILS) signal. In recent years, researchers have started incorporating vectoring strategies into the sequencing of arrival flights to absorb delays. Turning legs and parallel legs were designed to achieve route stretching, but they only offer a limited set of path options [6]. Imen et al. [7] extended this research by providing turning legs with more options for flexibility. The advantage of vectoring is that it allows for flexible trajectory changes through the combination of multiple heading instructions. However, the drawback is that it significantly increases the task load for ATCOs. Another strategy is to ask the flight to fly directly to a specified point (usually a point on the STAR), e.g. Direct to IAF, achieving a route shortening by bypassing certain segments of the STAR. Allowing for direct routes from the intermediate waypoints improves the overall arrival performance, including fuel savings and reduction of gaseous and particle emissions [8]. In addition, the ATCOs only need to issue a single instruction, leading to a lower task load compared to the continuous demands of issuing vectoring instructions.

It is important to note that these works focus on developing the optimal routes in the pre-tactical phase, rather than making real-time trajectory intervention decisions. The crucial decision here is to find the optimal timing for arrival flight to deviate from the STAR while considering the stochastic nature of flight operations. To handle real-time decision-making problems with stochasticity, Reinforcement Learning (RL) is a promising candidate, as it has already achieved impressive results in areas such as Conflict Resolution, Departure Slotting and Pushback Rate Control. The authors in [9] used RL to suggest departure slots while considering operational constraints and uncertainties. One research for conflict resolution is [10] in which the RL is utilized to handle the flight uncertainty and performance challenges. In order to manage taxi delay under uncertainties, the authors in [11] proposed a RL-based approach utilizing pushback rate as agent's action. These works indicate that RL has great potential in solving sequential decision-making problems in Air Traffic Management (ATM).

To the best of our knowledge, this is the first attempt to develop a Reinforcement Learning Model to make real-time direct-to IAF decision. The contributions of this paper are as follows:

- A framework for solving the real-time direct-to IAF problem using Reinforcement Learning model is proposed. The approach can be easily adopted to any other TMAs with limited calibrations, offering flexibility and scalability across different TMAs.
- A typical TMA environment for a multi-airport system has been established, which includes arrival operations for airports located within the TMA, as well as overflying operations for flights crossing the TMA to land at neighboring airports.
- This approach can handle the challenge of reducing arrival transit time considering the convergence of arrival flights from different directions, as well as the potential interactions between arrival flights and overflying flights.

II. Overview

In a multi-airport system, arrival operations within the TMA need to simultaneously consider the convergence of flights heading to the same airport and the crossing of flights heading to different airports. This makes it more challenging compared to single-airport scenarios. The target of this work focuses on supporting Approach Controllers in real-time making decisions for arrival flights to directly fly to IAF in a multi-airport environment. The overview of the RL-based approach proposed in this paper is illustrated by Figure 1.

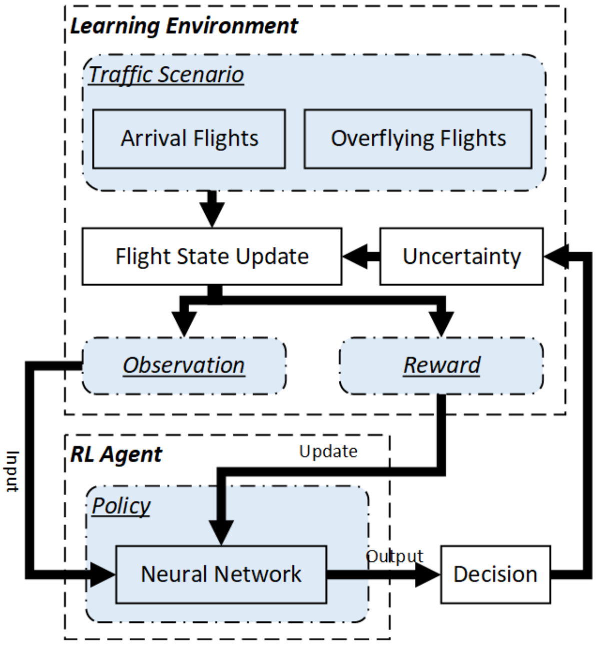


Figure 1. The diagram of the proposed RL-based approach for direct-to IAF decision making problem. There are two main modules: the Learning Environment and the RL Agent. The Learning Environment is primarily responsible for generating traffic scenarios, updating flight statuses, providing the agent with the necessary observations, and calculating rewards based on the agent's actions. The RL Agent mainly consists of a policy, which makes decisions based on observations from the Learning Environment and receives rewards from the Learning Environment to update its policy.

In this initial exploratory phase, this work focuses on making direct-to IAF decisions for arrival flights coming from a single direction. The detailed problem will be discussed

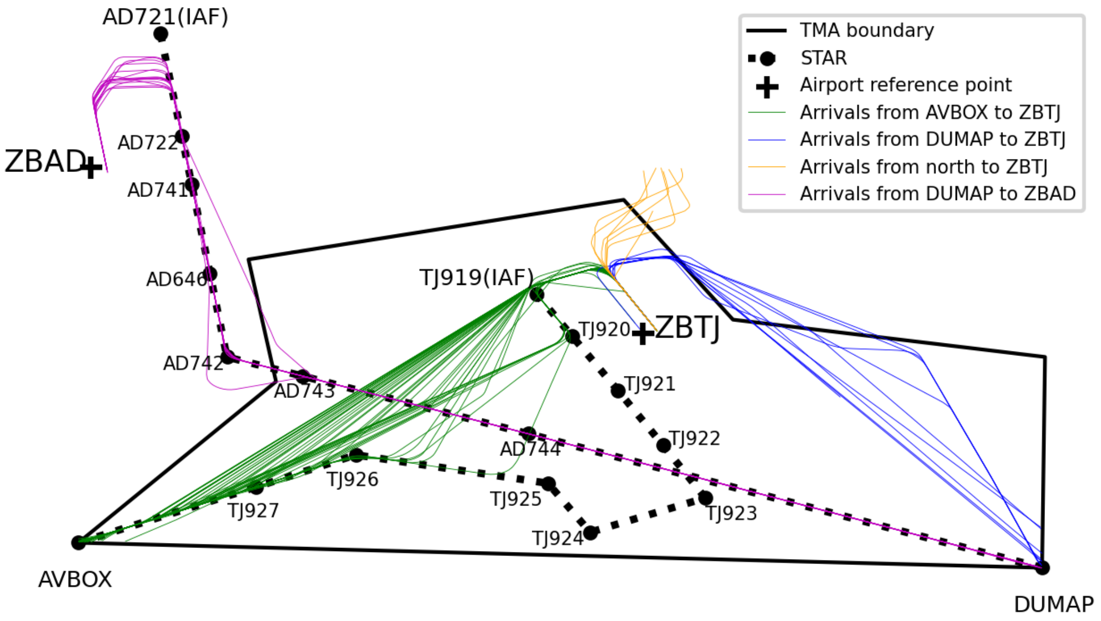


Figure 2. A TMA in a Multi-Airport System in Northern China, with Tianjin Binhai International Airport (ZBTJ) and Beijing Daxing International Airport (ZBAD) nearby. To focus on the problem addressed in this work, only two STARs are depicted in the figure: one for ZBTJ (named AVBOX-4ZA) and the other for ZBAD (named DUMAP-11A). The complete route for AVBOX-4ZA is: AVBOX-TJ927-TJ926-TJ925-TJ924-TJ923-TJ922-TJ921-TJ920-TJ919 (IAF), and the complete route for DUMAP-11A is: DUMAP-AD744-AD743-AD742-AD646-AD741-AD722-AD721 (IAF). There are also actual arrival flight trajectories on May 1, 2024, for all three directions to ZBTJ during southbound operations (shown in green, orange, and blue) and the actual flight trajectories for arrivals to ZBAD from DUMAP during southbound operations (shown in magenta).

in section III. The RL agent takes observation of the environment as input, processes them through a neural network, and outputs a decision on whether to execute a direct-to IAF routing considering the uncertainty arising from delays in decision execution. That decision is used to modify the current flight state and its quality is also evaluated through a reward mechanism. Action space and three main components of the learning environment, i.e., traffic generation, observation space and reward mechanism, will be described in details in section IV. During the training phase, rewards are used to update the parameters of the neural network. The RL algorithm, Proximal Policy Optimization (PPO), is adopted to train the agent, which is described in section V, followed by the detailed setting of experiments in section VI.

III. Problem Description

In TMA, direct-to is a commonly used control strategy by ATCO for arrival flights. It can accelerate air traffic flow and reduce transit time before landing, thereby lowering fuel consumption and pollutant emissions, as well as alleviating their own workload for monitoring the traffic. In existing studies, whether it's regarding route stretching [6], [7] or route shortening [8], it is assumed that flights can only begin to implement these strategies when passing through designated waypoints on the STAR. However, in actual operations, ATCO can issue instructions to change a flight's trajectory at any position. Figure 2 depicts a TMA in a multi-airport system, containing part of the STARs and actual arrival flight trajectories. It can be observed from Figure 2 that no flight fully adhered to the STAR, i.e., AVBOX-4ZA, throughout the day. Instead, all flights follow the direct-to TJ920/IAF routes. Additionally, flights did not start executing direct-to instruction only at intermediate waypoints of the STAR but could do so from any position.

Although direct-to offers the advantages described above, ATCOs must consider multiple factors when making decisions, making this task challenging in terms of balancing safety and efficiency. The two main factors are as follows:

- The convergence of arrival flights from different directions. ATCOs need to estimate the time at which a flight executing a direct-to instruction will arrive at the convergence point on final approach segment, ensuring it meets the required horizontal separation with preceding and following aircraft, which may be coming from other directions. Figure 2 shows the flight trajectories approaching ZBTJ from different directions during southbound operations on May 1, 2024. All trajectories eventually converge near the Intermediate Fix (IF) on the final approach leg.
- The potential interactions of trajectories with flights head-

ing to nearby airports. ATCOs need to estimate whether the trajectory of a flight executing a direct-to instruction maintains the required safety separation from crossing flights at their closest point. Figure 2 also depicts the flight trajectories from DUMAP to ZBAD. It is evident that these trajectories inevitably intersect with those of flights from AVBOX to ZBTJ. If direct routes are used, the separation between flights must be entirely ensured by the ATCO.

Based on the above analysis, this work aims to address the real-time decision-making problem of directing arrival flights to the IAF. This problem can be described as a Markov Decision Process (MDP), where decisions on whether a flight should fly directly to the IAF are made at each time step, ultimately forming a discrete sequence of decisions. At the initial stage of solving this complex problem, the study begins by focusing on controlling arrival flights in a single direction.

IV. Learning Environment for Decision Making

Traffic generation is to create diverse traffic scenarios, allowing agent to fully explore and learn. Observation is the only information that agent can obtain about the environment it need to learn, and based on this, it make decision and receive feedback from the environment, which is the reward.

A. *Traffic generation*

This work is based on the southbound operations of ZBTJ, constructing a Tianjin TMA simulation environment. As shown in Figure 2, there are three ZBTJ arrival traffic flows and one ZBAD arrival traffic flow within this TMA. This work requires generating three types of traffic:

- The arrival flight from AVBOX to ZBTJ, which is controlled by the agent. As overtaking in TMA is typically not allowed, this work assumes that at each timestep, the ATCO only needs to decide whether the first arrival flight in the queue can proceed directly. Thus, in each scenario, only one controlled flight entering the TMA from AVBOX needs to be generated. The TMA entry time and Wake Turbulence Category (WTC) of the flight are generated randomly.
- The arrival flights from other TMA entry points to ZBTJ, which are not controlled by the agent. In each scenario, the Estimated Time of Arrival (ETA) and WTC of each flight are randomly generated, and the separation requirements between flights are ensured. The number of flights to be generated is equal to the maximum number of flights recorded in these directions historically.
- The arrival flight from DUMAP to ZBAD, which is not controlled by the agent as well. According to statistical analysis of historical data, at most one arrival flight from DUMAP to ZBAD exists within the Tianjin TMA at any given time. Therefore, in this work, only one overflying flight is generated with random WTC in each scenario, and the flight enters the TMA at the first timestep.

B. *Observation space*

Observation is all the information that an agent can acquire about the environment. Theoretically, the more complete the observation, the more accurately it can describe the state of the environment. However, this will result in a high-dimensional observation space, making it more difficult for the agent to learn. Therefore, most studies adopt local observations as a substitute for global observations, which is also the approach taken in this paper.

When executing a direct-to action, the agent needs to know three types of information: 1) In the landing queue sorted in ascending order of ETA, the difference in ETA between the controlled arrival flight and the flight ahead of it (x_a), as well as the difference in ETA with the flight behind it (x_b); 2) The distance between its remaining trajectory and the overflying flight's trajectory at the closest point (d); 3) Considering the delay in pilots executing a direct-to instruction, a forward-looking time (u) needs to be factored in. As a result, the observation is a vector with fixed length of three: $O = [O_a, O_b, O_o]$, and each element is calculated as follows:

$$O_a = (\sum_{i=1}^{u} \mathrm{sign}(x_a^i - s))/u \tag{1}$$

$$O_b = (\sum_{i=1}^{u} \mathrm{sign}(x_b^i - s))/u \tag{2}$$

$$O_o = (\sum_{i=1}^{u} \mathrm{sign}(d^i - s_r))/u \tag{3}$$

Equations (1) to (3) express the probability of conflicts when executing a direct flight action with respect to the preceding aircraft, following aircraft, and crossing flights under uncertainty. The values of these three equations range from $[-1, 1]$. A value of 1 indicates that there are no conflicts under any circumstances, while a value of -1 indicates that there are conflicts under all circumstances. x_a^i (x_b^i) represents the difference in ETA between the controlled flight and the preceding (following) one when there is a delay in execution of $i \in [1, 2, ..., u]$. s is the maximum of the wake turbulence separation s_w and the radar separation s_r, and it is converted into a time-based separation according to the aircraft's ground speed v, as shown in Equation (4). The sign function, denoted as sign(x), is defined such that its value is 1 when $x \geq 0$, and -1 otherwise, as shown in Equation (5).

$$s = \max(s_w, s_r)/v \tag{4}$$

$$\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{5}$$

The embedded information ensure the agent have a clear understanding of the conflict probability between the controlled arrival flight and other flights after executing a direct-to maneuver.

C. Action space

When the controlled arrival flight enters the TMA, it becomes an active flight. At each time step, the agent needs to decide whether the active flight should continue to follow the STAR (F) or proceed directly to the IAF (D). If the 'F' action is selected, the active flight will continue to follow the STAR until the next time step. This procedure is continued until the 'D' action is selected or reach the terminated state, i.e., the flight arrives at TJ919. As shown in Figure 2, once the flight reaches TJ919, selecting 'F' or 'D' has the same effect.

D. Reward Mechanism

Safety is the primary consideration in ATM decision-making. Since the goal of ATM is to improve efficiency while ensuring safety, the reward mechanism is based on both safety and efficiency. As previously stated, separations must be maintained between controlled arrival flight and other arrival flights, as well as between controlled arrival flight and overflying flight. As a consequence, the reward mechanism is designed in such a manner that it guides the model to achieve a shorter flight path while eliminating any possible dangers associated with violating the safety separations. Every time step, the agent will receive a penalty that encourages making the direct-to IAF decision as early as possible. After the agent makes a direct-to IAF decision or reaches the terminated state, it will receive an additional reward based on whether the separations are satisfied. If the active flight satisfies all required separations, an additional reward of $+1$ will be given; otherwise, a penalty of -1 will be imposed. The total reward for one episode of the agent is the accumulation of all time step's reward, as shown in Equation (6):

$$R = \sum_{t=1}^{n} (p_t) + r \tag{6}$$

where n denotes time steps in one episode; p_t is the time step penalty; r represents whether the direct-to IAF action violates separations.

V. Reinforcement Learning Approach

Proximal Policy Optimization (PPO) [12] is a widely-used reinforcement learning algorithm, primarily introduced to address stability and efficiency challenges in policy gradient methods. PPO belongs to the class of actor-critic algorithms, where the actor updates the policy and the critic evaluates it by estimating value functions.

One of the key innovations of PPO is the use of a clipped objective function, which ensures that the new policy does not deviate too much from the old policy. The core objective function in PPO is given by:

$$L^C(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right] \tag{7}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio between the current and previous policies, and $\hat{A}_t$ is the estimated advantage function. The term ϵ controls the allowed deviation from the old policy, and the clipping function prevents excessively large policy updates.

PPO also uses a critic network to estimate the value function, and the critic is updated by minimizing the squared error between the predicted value and the actual return. This is represented as:

$$L^V(\theta) = \frac{1}{2} \mathbb{E}_t \left[(V_\theta(s_t) - V_t^{\text{target}})^2 \right] \tag{8}$$

The final objective combines the clipped policy loss and the value function loss, along with an entropy bonus to encourage policy exploration:

$$L(\theta) = L^C(\theta) - c_1 L^V(\theta) + c_2 \mathbb{E}_t \left[\mathcal{H}[\pi_\theta](s_t) \right] \tag{9}$$

where c_1 and c_2 are coefficients balancing the losses, and $\mathcal{H}[\pi_\theta](s_t)$ represents the entropy of the policy, which encourages exploration by discouraging premature convergence to suboptimal deterministic policies.

PPO strikes a balance between the simplicity of policy gradient methods and the stability of trust-region methods, making it a preferred choice in many deep reinforcement learning applications. This work adopts the PPO algorithm to train the agent's policy model for the environment described in section IV. The hyper-parameters will be listed in section VI.

VI. Experimental Setting

The environment in this study is constructed following the OpenAI Gymnasium [13] framework, and the PPO algorithm implementation is based on the Stable-Baselines3 [14] reinforcement learning library. The hyper-parameter settings for PPO can be found in Table I. The policy network and value network in PPO are designed with a similar architecture: two hidden layers, each with 32 neurons, and the Rectified Linear Unit (ReLU) activation function.

TABLE I. PPO Parameters

Parameters	Values
Training Iterations	1e+06
Number of Parallel Environments	1
Discount Factor	0.999
Learning rate	0.0003
Buffer size	1024
Batch size	64
c_1	0.5
c_2	0
ϵ	0.2
Optimizer	Adam

Table II shows the set of parameters which are used in the environment. In the actual operations of Tianjin TMA, there are no light aircraft, so this work only considers medium and heavy aircraft when randomly generating traffic scenarios. When a heavy aircraft is in front and a medium aircraft is behind, the required wake turbulence separation is 5 nautical miles. In other situations, there is no wake turbulence separation requirement between consecutive landing flights, but when converging on the final approach, a 3-nautical-mile radar separation must be maintained. This work generates 5 arrival flights from other directions in each scenario, which is the maximum number of flights recorded in these directions within 30 minutes historically. Based on historical data statistics, the controlled arrival flight's speed entering the TMA is set to 280 knots and its speed upon reaching the IF is set to 180 knots. For the overflying flight, the speed entering the TMA is set to 340 knots, and the speed leaving the TMA is set to 270 knots. This work assumes that the flights follow uniform deceleration motion.

TABLE II. ENVIRONMENT PARAMETERS

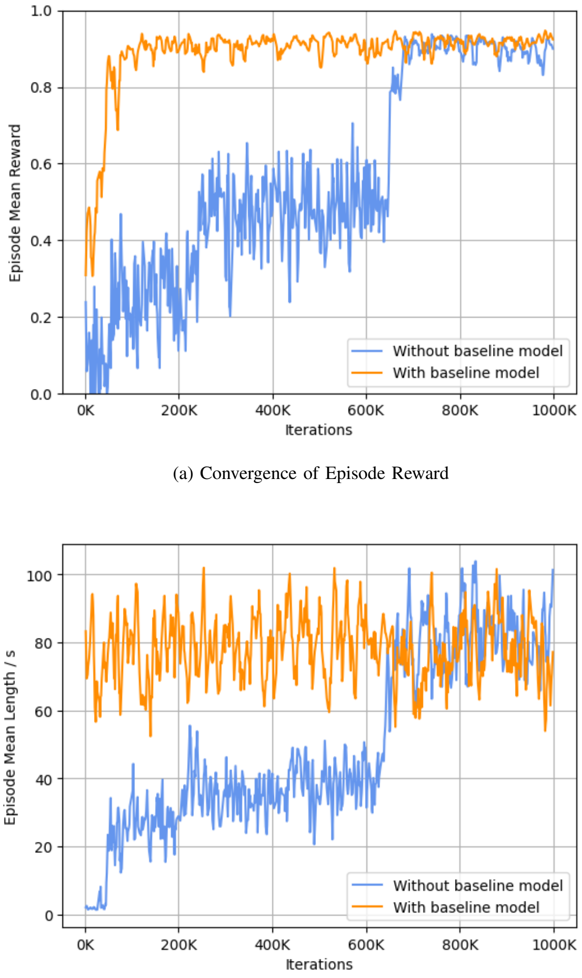
Parameters	Values
WTC Distribution	{H:0.5, M:0.5}
s_w	5 nm (M follows H) and 0 (otherwise)
s_r	3 nm
v	180 kts
u	10 s
r	1 (no conflict) and -1 (otherwise)
p_t	-0.001

The larger the agent's forward-looking time, the larger the observation space, which increases the difficulty of learning the policy. This paper considers a moderate forward-looking time of 10 seconds. After the agent makes a direct-to decision based on its observation, it will receive a random delay in execution. During this period, the activated flight will continue to follow the same acceleration along the STAR. Subsequently, the flight will adopt a new acceleration to fly directly to the IAF, calculated based on the current speed, the final speed, and the remaining distance. This work does not consider the impact of wind and ignores trajectory changes during the turning process. This paper considers a maximum execution delay of 5 seconds and assumes that the delay follows a discrete uniform distribution, with possible values of 0s, 1s, 2s, 3s, 4s, and 5s.

VII. Results and Discussions

The model training in this work was conducted using a standard Windows laptop with CPU i7-12700H. Figure 3 shows the convergence of the training process (the blue line). After approximately 700,000 iterations (3.4 hours), the model began to exhibit stable performance. The average episode reward stabilized around 0.9, and the average episode length remained stable between 60 s and 100 s. It is worth noting that in the early stages of training, the model's performance remained at a relatively low level, with the average reward fluctuating around 0.5. This work explores a method to accelerate training process. First, the model is pre-trained in a simple environment without uncertainties until convergence, obtaining a baseline model. Then, using the baseline model as a foundation, training continues in a more complex environment with uncertainties. The baseline model quickly converged during training, reaching an average reward of around 0.93 within 80,000 iterations (0.3 hours), as shown in Figure 4.

The training process based on the baseline model in the stochastic environment is represented by the orange line in Figure 3. It exhibits relatively high performance from the start, with an average reward of around 0.4, which is already comparable to the model's performance after 200,000 iterations (1.4 hours) as shown by the blue line. Subsequently, within 100,000 iterations (0.3 hours), the average episode reward rapidly increased to around 0.9, which is comparable to the model's performance after 700,000 iterations (3.4 hours) as shown by the blue line. The episode length reflects the time from when a flight enters the TMA to when the agent makes a direct-to decision for it. In the environment described in this paper, the minimum value of the episode length is 1 s, which indicates that a direct-to decision is made for the flight at AVBOX. The maximum value of the episode length is 436 s, which means the flight follows the STAR all the way to TJ923. In both training modes, the episode mean length eventually fluctuates

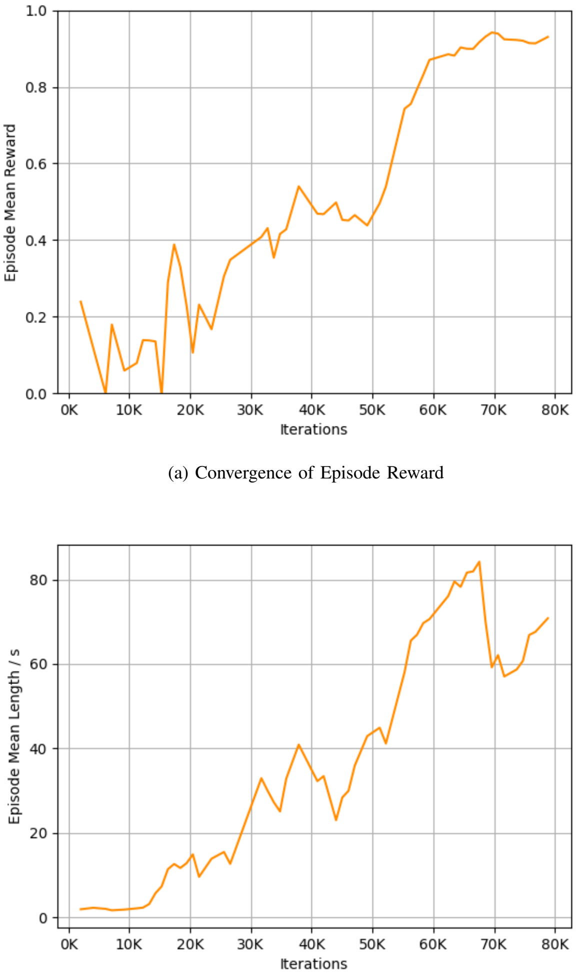


(a) Convergence of Episode Reward

(b) Convergence of Episode Length

Figure 3. Illustration for the convergence of the training process in a stochastic environment. The blue line represents the training process of the agent's policy starting from randomly initialized neural network parameters, while the orange line represents the training process starting from a model that has converged in the deterministic environment (the model after 80,000 iterations in Figure 4).

around 80 seconds, indicating that, in this environment, the average time a flight follows the STAR is 80 seconds, after which it receives a direct-to-IAF instruction. Ultimately, the new training method reduced the training time by 82% without sacrificing model performance. These experimental results can provide valuable insights for addressing reinforcement learning problems in complex environments.



(a) Convergence of Episode Reward

(b) Convergence of Episode Length

Figure 4. Illustration for the convergence of the training process in the deterministic environment.

To further evaluate the models, 10000 episodes were set up to evaluate the two trained models, i.e., Without baseline model (WO) and With baseline model (WB), and the ad-hoc model in identical environments. For basic arrival operations, if direct-to IAF instruction will not cause any conflicts, the ATCO will allow the flight to proceed directly. When making decisions, ATCOs typically add a buffer on top of the minimum separation standards to account for various uncertainties. However, real-world uncertainties are difficult to predict, so this work assumes that the agent cannot foresee environmental uncertainties. To ensure fairness in comparison, this work assumes that in ad-hoc operations, the controller can only obtain information about whether there is a conflict with other flights based on the projected direct flight path.

The number of conflicts occurring within 10,000 episodes and the average TMA transit time of arrival flights are shown in Table III. With similar TMA transit times, the model trained in this work achieves completely conflict-free operations, while the Ad-hoc model experiences 70 conflicts. This demonstrates that the trained models in this work possess a good functionality in safety management. On the other hand, the test results of WO and WB are completely consistent, which demonstrates that the accelerated training method explored in this study does not compromise model performance. This method shows great potential for application in handling more complex air traffic decision-making problems.

TABLE III. Comparison of Average Trained Models' Performance over the Ad-hoc Model

	Conflict number	TMA transit time (s)
Ad-hoc	70	449
WO	0	451
WB	0	451

Figure 5 shows the direct flight path decided by WB over 1000 randomly generated episodes. From Figure 5, it can be observed that the decision timing of the RL model slightly differs from the actual trajectory shown in Figure 2. The main reasons for this discrepancy are likely twofold: 1) The actual trajectory reflects two different types of direct routes, i.e., direct flight to IAF and direct flight to TJ920, whereas the action space designed for the agent in this work does not include the decision for direct flight to TJ920. 2) The traffic scenarios in this work are randomly generated, which may include scenarios that have not occurred in historical data. Theoretically, any waypoint on the STAR can be regarded as a direct-to target. However, the goal of this study is to propose a generalizable RL-based direct-to decision model, so we do not list all possible cases. Users only need to make minor adjustments to the model proposed in this paper to apply it to different TMAs and STARs.

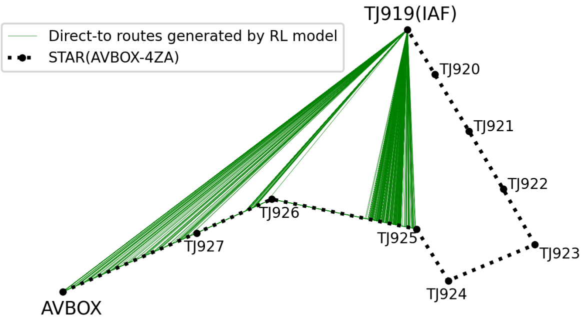


Figure 5. Direct routes generated by RL model.

VIII. Conclusion

This research proposes a Reinforcement Learning approach for real-time arrival direct-to decision making in multi-airport system. The approach takes into account the convergence of arrival flights at the same airport, the crossing of arrival flights from neighboring airports, and the uncertainty in pilots' execution of ATCO's decisions. A method for accelerating reinforcement learning training is explored, using a model that converges quickly in a simple environment as the foundation, and further training it in a more complex environment. A case study for Tianjin TMA is developed and investigated. The results show that the accelerated training method can significantly reduce training time without sacrificing the model's convergence performance. The trained model performs comparably to the Ad-hoc model in terms of TMA transit time, but it is able to achieve conflict-free operations, something the Ad-hoc model cannot accomplish. The results demonstrated the potentials of AI systems as decision support tools in the field of ATM.

In the next step, flights from different directions need to be considered in a coordinated manner. More ATCO intervention methods, such as vectoring and speed control, should be added to the agent's action space to enhance flexibility. Additionally, more types of uncertainties need to be taken into account to increase the model's potential for real-world application.

Acknowledgment

This research is supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No.KYCX24_0598), the China Scholarship Council (No.202306830150) and the National Natural Science Foundation of China (No.62076126). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Jiangsu Education Department, the China Scholarship Council and the National Natural Science Foundation of China.

References

[1] International Civil Aviation Organization (ICAO), *Procedures for Air Navigation Services – Aircraft Traffic Management (Doc 4444)*, sixteenth ed., 2016.

[2] J. Ma, D. Delahaye, M. Sbihi, and M. Mongeau, "Integrated optimization of terminal manoeuvring area and airport," in *6th SESAR Innovation Days*, 2016.

[3] C. Yin, W. Zeng, H. Jiang, X. Tan, and W. Tian, "Standard procedure-guided flight trajectory pattern mining for airport terminal airspace," *International Journal of Aeronautical and Space Sciences*, 2024.

[4] M. C. R. Murça, "Identification and prediction of urban airspace availability for emerging air mobility operations," *Transportation Research Part C: Emerging Technologies*, vol. 131, p. 103274, 2021.

[5] M. C. R. Murça and R. J. Hansman, "Identification, characterization, and prediction of traffic flow patterns in multi-airport systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, pp. 1683–1696, 2019.

[6] D. Gui, M. Le, Z. Huang, J. Zhang, and A. D'Ariano, "Optimal aircraft arrival scheduling with continuous descent operations in busy terminal maneuvering areas," *Journal of Air Transport Management*, vol. 107, p. 102344, 2023.

[7] I. Dhief, M. Feroskhan, S. Alam, N. Lilith, and D. Delahaye, "Meta-heuristics approach for arrival sequencing and delay absorption through automated vectoring," in *IEEE Congress on Evolutionary Computation*, 2023.

[8] H. Hardell, T. Polishchuk, and L. Smetanov´a, "Arrival optimization with point merge in a dual-runway environment," in *13th SESAR Innovation Days*, 2023.

[9] D.-T. Pham, L. L. Chan, S. Alam, and R. Koelle, "Real-time departure slotting in mixed-mode operations using deep reinforcement learning : a case study of zurich airport," in *14 USA/Europe Air Traffic Management Research and Development Seminar*, 2021.

[10] Y. Chen, M. Hu, L. Yang, Y. Xu, and H. Xie, "General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution," *Transportation Research Part C: Emerging Technologies*, vol. 151, p. 104125, 2023.

[11] H. Ali, D.-T. Pham, and S. Alam, "Toward greener and sustainable airside operations: A deep reinforcement learning approach to pushback rate control for mixed-mode runways," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2024.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, *et al.*, "Gymnasium: A standard interface for reinforcement learning environments," *arXiv preprint arXiv:2407.17032*, 2024.

[14] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.