Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)

Real-time Departure Slotting in Mixed-Mode Operations using Deep Reinforcement Learning: A Case study of Zurich Airport

Duc-Thinh Pham, Li Long Chan and Sameer Alam
Air Traffic Management Research Institute,
School of Mechanical and Aerospace Engineering
Nanyang Technological University, Singapore
Email: {dtpham | lilong.chan | sameeralam}@ntu.edu.sg

Rainer Koelle
Operational ANS Performance
Performance Review Unit
EUROCONTROL
Email: rainer.koelle@eurocontrol.int

***Abstract*—A mixed-mode runway operation increases the runway capacity by allowing simultaneous arrival and departure operations on the same runway. However, this requires careful evaluation of safe separation by experienced Air Traffic Controllers (ATCOs). In daily operation, ATCOs need to make real-time decisions for departure slotting. However, an increase in runway capacity is not always guaranteed due to the stochastic nature of arrivals and departures and associated environmental parameters. To support ATCOs in making real-time departure slotting decisions, this paper proposes a Deep Reinforcement Learning approach to suggest departure slots within an incoming stream of arrivals while considering operational constraints and uncertainties. In this work, novel state representation and reward mechanism are designed to facilitate the learning process. Experimentation on A-SMGCS data from Zurich airport shows that the proposed approach achieves an efficiency ratio of more than 83.8% of the expected runway capacity while maintaining safe separation distances in mixed-mode operations. The results of this work have demonstrated the potentials of Deep Reinforcement Learning in solving decision-making problems in Air Traffic Management.**

***Index Terms*—Airport Runway Control, Reinforcement Learning, Air Traffic Control, departure sequencing, departure slotting**

I. Introduction

Airports, with their fixed infrastructure, have emerged as main bottleneck in the air transportation system [1]. They are facing the capacity challenge with prolonged congestion and delay problems [2]. In which, runway capacity is one major constraint that may pose a challenge to meet the future air travel demand. Since runways extensions are costly and often challenging, operational solutions to increase runway capacity are gaining more and more attention. According to ICAO's Manual On Simultaneous Operations On Parallel Or Near-Parallel Instrument Runways, an airport with parallel or near parallel runways has 2 general modes of operations: (1) Segregated Mode (Each runway is used strictly for Departures or Arrivals only), (2) Mixed Mode (Runways can serve both Departures and Arrivals Simultaneously) [3]. The mixed-mode is a promising approach to increase runway capacity. For example, the work, done in [4], shows an increase in runway capacity under mixed-mode operations. Mixed-mode is more efficient since alternating take-offs and landings on the runway can effectively reduce delays due to wake vortex constraints [5]. Moreover, with the current impact of COVID-19, in single runway operation is also an considerable option to reduce operational cost for airport with less traffic demand. From the end of last year 2020, Heathrow Airport, UK, has consolidated their operations and returned to single runway using mixed mode operations [6]. However, operating in mix-mode is also challenging, any mismanagement of would increase the probability of missed approach, which may potentially lead to a reduction in runway capacity as compared to operating in segregated mode [3].

Mixed-mode runway operations require the sequencing of arrivals and departures, referred to as Aircraft Landing Problem (ALP) or the Aircraft Take-Off Problem (ATP) in the literature [7]. This sequencing problem in the ALP/ATP is typically solved in the pre-tactical phase and can be shown to be a variant of the Job-Scheduling Problem (JSP) which is NP-Hard [8], [9]. This would mean that an optimal solution to sequencing might be impossible to find in a reasonable time and other heuristic/approximate methods are used in the literature [10]–[12]. For example, [13] studied the use of Mixed-Integer Programming to optimally schedule arrival and departure sequence on a mixed mode runway in a practical timing by iterating on a previously known schedule (known as "Warmstart") instead of iterating on a new a random schedule. The solutions from solving ATP/ALP using approximate methods may not be strictly followed in practice due to the inherent stochasticity in the environment including uncer-

National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore.

tainties in aircraft arrival timings. This requires an ATCO's manual intervention to ensure the solution's feasibility [5], [14] which at times is hard to achieve. In order to eliminate uncertainties from taxi timings, [5] suggested to shift the scheduling of aircraft at the holding area of the runway. This method of scheduling only at the holding area would mean that calculations would have to be started and completed closer to the time when arrival aircraft lands. It is important to note that these works focus on developing the optimal sequence (arrangement) of aircraft rather than detecting and slotting departures in a mixed mode runway. The crucial decision here is to identify gaps in the landing stream while considering the stochastic nature of arrival and departure operations. Identifying arrival gaps (to schedule departures) in a real-time environment demands high cognitivity from air traffic controllers (ATCOs), which may be a source of increased workload thereby affecting flight safety.

To deal with the stochasticity, Reinforcement Learning (RL) is one promising candidate which has been explored recently. The authors in [15] used RL to predict departure taxi-out time. Behavioral Cloning and Inverse Reinforcement Learning is proposed as an potential solution for predicting ground delay program implementation decisions [16]. Another research focused on utilizing DRL for surface movement planning is [17] in which the model plans conflict-free paths while considering potential airside conflicts and take-off time constraints. All of these works share similar characteristics in balancing safety and optimal solutions. They also utilize the use of a simulator to test the efficacy of Reinforcement Learning models in solving Air Traffic Management problems.

To the best of our knowledge, this is the first attempt to develop a Reinforcement Learning Model in a Simulated Mixed-Mode Runway environment to identify and slot departures in a Mixed-Mode Runway. The contributions of this paper are as follows:

- A framework which can solve the real-time departure slotting problem using Deep Reinforcement Learning model is proposed. The approach can be easily adopted to any other airports with limited calibrations.
- A data-driven simulator that simulates runway operations for Zurich Airport Runway (Runway 28) is built. The simulator has the ability to generate a wide range of realistic scenarios, novel state representation and reward structure which are designed for effectively training the model.
- This approach can handle the challenge of balancing multiple departure queues considering different aircraft' wake turbulence category (WTC) and uncertainties in runway occupancy time (ROT) and delays in decision execution (ATC-Pilot).

II. Overview

As the target of this work focuses on supporting Tower Controllers in real-time handling or slotting departures, the interaction or coordination between departure and arrival controllers will not be considered. As a result, arrival sequences are used as input and kept untouched during the decision process. The departures for multiple queues are generated with random uniform intervals and aircraft' categories. The overview of the approach proposed in this paper is illustrated by Figure 1. Even though historical data is used to estimated several parameters of the proposed framework, one of its main purpose is to create traffic scenarios in the learning environment. The traffic scenarios, including the arrivals and departures, will be standardized and presented in an appropriated form (images) called state representation. Based on observing current state, a Reinforcement Learning agent will make the slotting decision considering uncertainties such as variations of arrivals' ROT. That decision is used to modify the current runway status and its "quality" is also evaluated. Those three main components, e.g., historical data, learning environment and Reinforcement Learning agent, will be described in details in following sections.

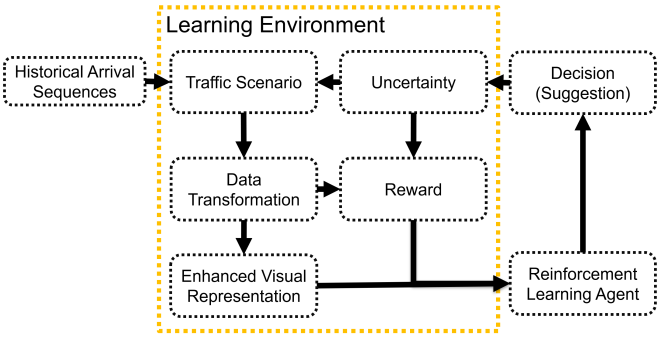


Figure 1: The diagram of the proposed approach for departure slotting problem.

The trained model can be used in either autonomous or manual mode. In the former, the model will make decision for a departure slot (and associated arrival sequencing / runway utilisation). This decision will be communicated to pilot as a take-off clearance command. In the latter, the model's output is considered as a suggestion for ATCOs which is the ultimate goal of this work. Based on the suggestion, ATCOs can make his/her own decision for departure slot sequencing. while the model keeps monitoring the situation to update its suggestion. However, in this work, for the purpose of training and evaluation, the model operated in autonomous mode in which its decision is applied to control the departure aircraft. After the reception of the departure slot decision, the depart aircraft will perform its take-off procedure. The model's performance for different metrics are recorded and analysed to comprehend the model's pros and cons.

III. Zurich Airport (LSZH) A-SMGCS Data

In this study, one month (October 2019) traffic data of Zurich Airport (Figure 2), extracted from A-SMGCS system, is used for training and testing the proposed model. Zurich Airport has three runways: 16/34 of 3,700 m (12,100 ft) in

length, 14/32 of 3,300 m (10,800 ft) in length, and 10/28 of 2,500 m (8,200 ft) in length. For most of the day and in most conditions, runway 14 is used for landings and runways 16 and 28 are used for takeoffs, although different patterns are used early morning and in the evenings. The obtained dataset includes all trajectories (60828 trajectories) within the range of 32 NM from the airport. Since only part of the given dataset are flights which are take-off or landing at Zurich Airport, some processing steps are conducted to filler out noisy or irrelevant trajectories.

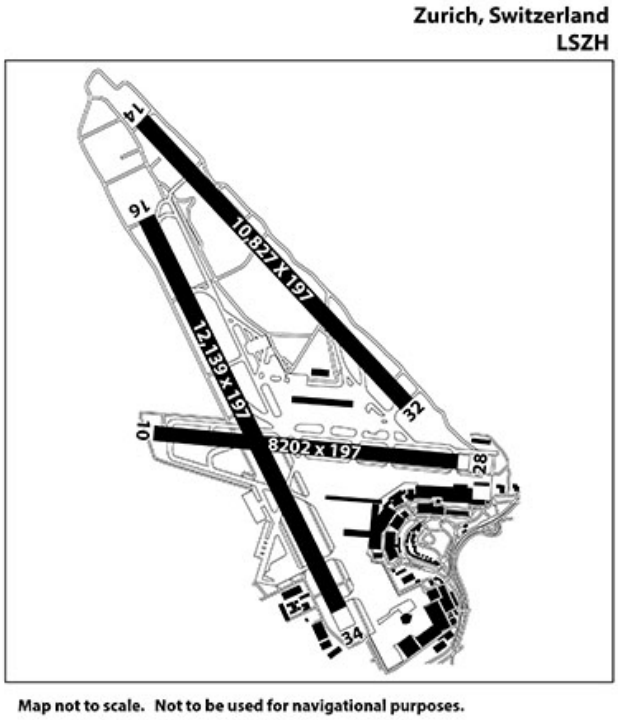


Figure 2: Zurich Airport has a complex runway system. Runway 10/28 crosses runway 16/34. The location of runway 28, between the north and south aprons, is also critical.

Cleaning of Data: Extracting Departures and Arrivals

As mentioned, the given dataset contains some noisy or irrelevant trajectories which need to be filtered out. Thus, the first processing step is to extract the take-off and landing trajectories from Zurich Airport data. However, since the information about flight types and runway usage isn't available in the obtained data, a simple algorithm is developed to extract that information. Polygons that enclose each of the three runways (Figure 3) are first used to isolate trajectories that intersects the runways for at least 30 consecutive points. Polygons that encloses areas at each of the runway's ends are then used to separate arrivals and departures by analyzing if the trajectory intersects the runway first. Trajectories that intersects the runway for at least 30 consecutive points and intersects the runway polygon first are departures. Trajectories that intersects the runway polygon for at least 30 consecutive points and intersects the polygon at the edge of the runway first are arrivals.

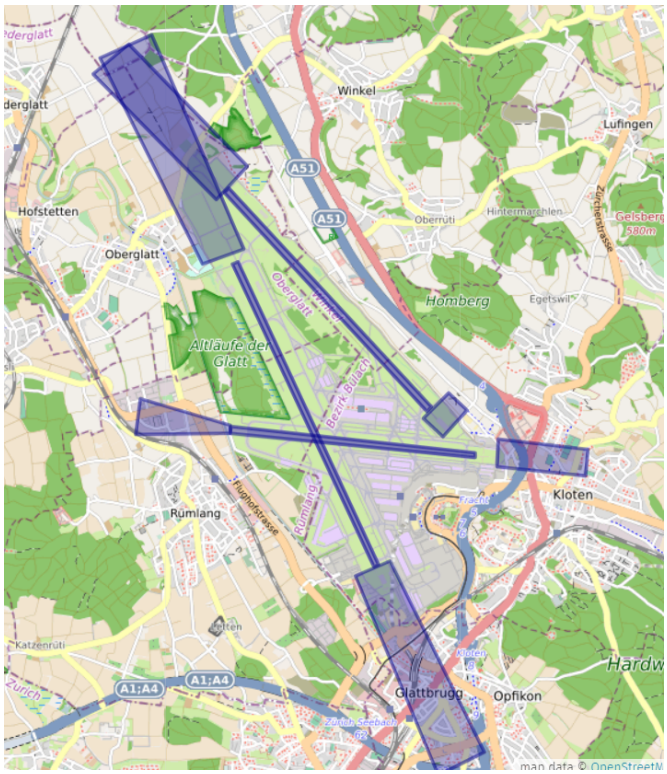


Figure 3: Polygons drawn in Zurich Airport Map. Each Polygon has vertices in Longitude and Latitude.

Runway Usage and Trajectories

After the cleaning step, the new dataset (19285 trajectories) is obtained and used for further investigation. Table I shows the summary of the runway usages in data, in which Runway 14/32 and 10/28 are major used comparing to Runway 16/34. Besides, Figure 4 provides a more detail information in terms of runway usages for takeoff and landing which confirmed the previous information about runway configuration of Zurich Airport. Runway 14/32 has more movements ($\approx 20\%$) than Runway 10/28 with significantly high number of arrivals.

TABLE I: Data Details

Description	Values
Date Range	02/10/19 to 30/10/19
Total Trajectories	60828
RWY 14/32 Trajectories	9282
RWY 16/34 Trajectories	2234
RWY 10/28 Trajectories	7769
Total Clean Trajectories	19285
Noisy Trajectories	41543

Mode of Operation at LSZH

The first timing at which an aircraft intersects the runway polygons can be considered to investigate the mode of runway operation at LSZH. Table II shows the percentage of flights

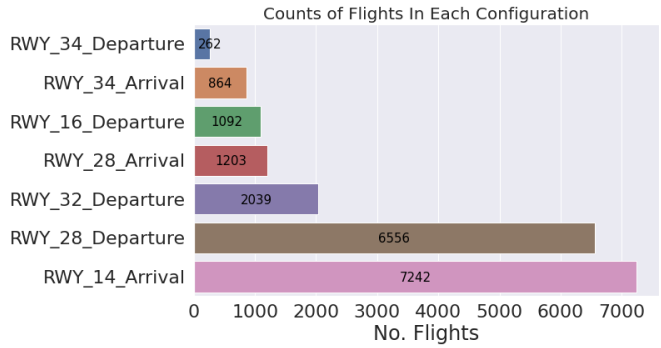


Figure 4: Number of Flights in each runway which Runway 14 has the maximum number of arrival.

in each runway where arrivals and departures are observed to be active within a 10 minutes interval.

TABLE II: Mixed Mode Operation Statistics

	Total Flights	Flights in Mixed Mode Operation
RWY 14/32	9282	1.35%
RWY 16/34	2234	4.476%
RWY 10/28	7769	0.811%

The low percentage of mixed mode operations suggests that Zurich Airport operates mainly in the segregated mode configuration. Moreover, this study require to select a runway which has sufficient gaps between arrivals and been able to operate takeoff and landing in same direction. Therefore, Runway 28 is selected in this study as the case study and only its corresponding data (e.g., 1203 landing trajectories and 6556 takeoff trajectories) is used in following steps. Noting that, the Runway 28's runway occupancy time (ROT) in selected data has the arrivals' ROT approximated by $\mathcal{N}(56.8, 10)$, and departures' ROT approximated by $\mathcal{N}(110, 53)$.

IV. Data-driven Simulator

A. Data Preprocessing

Figure 5 illustrates the arrival trajectories and the two focused views. In this work, as the input for the model, only part of trajectories, from the D10.3 point to the runway exit, are extracted and utilized. Even though, the considered distance of arrivals can be extended much longer, via conducted experiments, the selected range is appropriate for the scope of this work (e.g., real-time slotting decision at runway holding point) and sufficient for the model to converge. This distance may be extend in the authors' future work which considers the coordination between departures and arrivals. The extracted data is stored in a new dataset which is used for generating arrivals sequences in the simulator (describe in IV-B).

B. Traffic Scenarios

The simulator is constructed such that it can stream real arrival trajectories and represent them in a graphical format. During training, real trajectories are used and the frequency

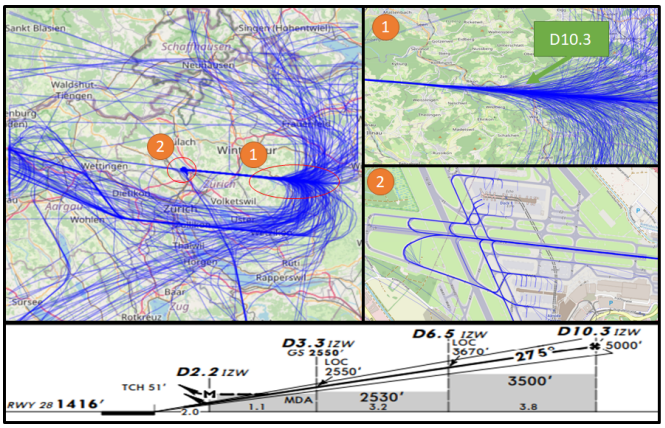


Figure 5: The visualizations arrival flights on LSZH runway 28. The considered range for arrivals is up to the merging point D10.3 (10.3NM from the Runway 28's threshold). (1) shows the closer view at the fixed point D10.3 where all the trajectories are merging, (2) shows the view at runway which illustrates the variations on selected runway exits.

of generating arrival aircraft is randomised (130s to 400s) to simulate various traffic density scenarios. This technique will improve the robustness of the agent by exposing it with several traffic conditions. The trained model is expected to behave similarly in real operations as the simulator is merely replaying real trajectories (with new visualization) during training and testing. In fact, real arrival sequences from data are also parts of testing scenarios.

The other element of the traffic scenario is departures. During the episode, departures are generated randomly with one out of three WTC (Heavy, Medium and Light) with similar probability. The generated departure will be also randomly assigned to a runway queue. If the length of the queue is larger than the pre-defined maximum queue length, that departure will be neglected (i.e not added to the queue). If it is added, it will join the queue with a given maximum waiting time T_w. This quantity, as part of the reward, will guide the model to balance between the waiting time of departures in queues. This departure generation process is designed to expose the model with different departure patterns during training. For evaluation, a real departure queues can be easily loaded into the simulator as the input scenario. Only two departure queues, located at the runway entry point of Runway 28, are considered in this study. The visualization of the scenario will be described in Section IV-E.

C. Slotting Decision

At each time step, the model will select one out of three decisions: Release aircraft from Queue 1 (R1), Release aircraft from Queue 2 (R2) or Hold both aircraft at the holding points (H).

When the decision is selected/recommended, if the confidence level of the algorithm is higher than the defined threshold ($\geq 50\%$), which is a tunable parameter, that decision will be executed. In case of releasing (R1 and R2), the selected

aircraft will start to move to the runway and then take-off. At the same time, aircraft in the selected queue will move to the next waiting position. In case of holding (H), all departures will standby and wait for the next time step.

To support the learning and avoid unnecessary risks, the releasing (R1 and R2) decision are only valid when the runway is clear from the previous departure.

D. Environment Uncertainties

There are two kinds of uncertainties in the simulation:

- Uncertainty of arrival sequences: because the real approach and landing profiles are used, their uncertainties such as landing speed, touch down time, runway exit and runway occupancy time are all preserved in A-SMGCS data. Besides, to simulate different traffic scenarios, the arrival gaps are randomly generated using uniform distribution. The parameters of the uniform distribution can be adjusted to control the traffic density level.
- Uncertainty of departures: (1) a random delays in the execution of the take-off decision is added to improve the realisticity of the simulation; (2) random departure ROT with distribution extracted from data; (3) during the running time, the departure for each queue is also generated randomly.

E. State Representation

Representing the spatial temporal data directly from sensor data is challenging as it requires a complex hand-crafted feature set to model the spatio-temporal information of the varying number of aircraft involved in the scenario (different number of aircraft can appear on the screen at different timings) while maintaining a fixed size vector as an input to the RL model. An image is an easy way to concisely and fully represent spatial temporal data. Figure 6 presents how the image of the scenario is constructed. As shown in the figure, the extremely long trajectories is divided into eight segments with similar distance. Then they are combined with bigger scale for segment 7 and 8 (the view of runway). This approach can concisely represent the final approach path and the runway scenario.

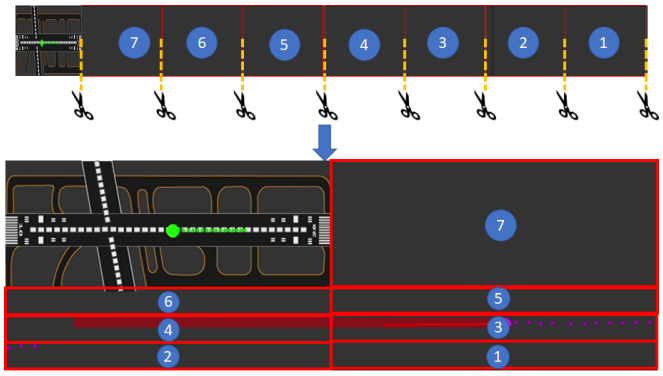


Figure 6: Constructing the image representation for both the final approach path and the runway scenario.

Moreover, to support for the deep learning framework, all necessary operational conditions and information must be visualized. Figure 7 shows the image with enhanced visualization. At each departure queue, departures are presented as circles with WTC letters and transition color to encode their waiting time in the queue. The Wake Vortex is presented as a blue region which decays over time. Before each arrival, the separation distance ($\lambda = 3NM$) is also presented as a red bar. Finally, a slotting signal is visualized as an indicator of model's decision in previous step.

In this work, the state representation is formed by stacking images with the size (64 X 200) across 4 timesteps. This approach can concisely represent the final approach path and the runway scenario.

F. Reward Mechanism

As mentioned in ICAO's Manual on Parallel Runway Operations [3], a gap that is sufficient to account for safety distances must exist between the incoming stream of arrivals such that a departure can utilise the runway for takeoff. When a departure clearance is given, the departure should ideally wait for the leading arrival to exit the runway, be at least more than the minimum arrival-departure distance away from the trailing arrival and not depart too closely with any leading departure. The list of major variables for slotting a departure in a mixed-mode runway is mentioned in Table III. An example of time-space diagram is presented in Figure 8 to demonstrate the relationship of mentioned variables with the movements. Red and Green lines represent successful arrivals and departures respectively. $Arrival_k$ (Orange) represents an arrival which violates the Departure-Arrival Separation δ. $Departure_k$ (Blue) represents a departure which violates the Departure-Departure separation ϵ_{ij}. Trajectories in this figure is shown with a clean solid line but however in reality the exact position in space and time is subjected to noise from sensors.

TABLE III: Stochastic Variables that affects mixed-mode runway operations [18]

Variable	Description
T_{ij}	Arrival-Arrival Separation Timing: Time between Leading Arrival of type i and Trailing Arrival of type j
$\frac{\delta}{V_j}$	Departure-Arrival Separation Timing: Separation Time between Leading Departure and Trailing Arrival of type j with velocity of V_j.δ represents the minimum arrival-departure separation
ROT_i	Runway Occupancy Time of Leading Arrival
ϵ_{ij}	Departure-Departure Separation Timing: Separation time between Leading departure of type i and Trailing departure of type j to account for Wake Vortexes.
n	Number of Departures that can fit into gap

Since safety is the most important aspect in ATM, the variables mentioned in TABLE III are considered in the

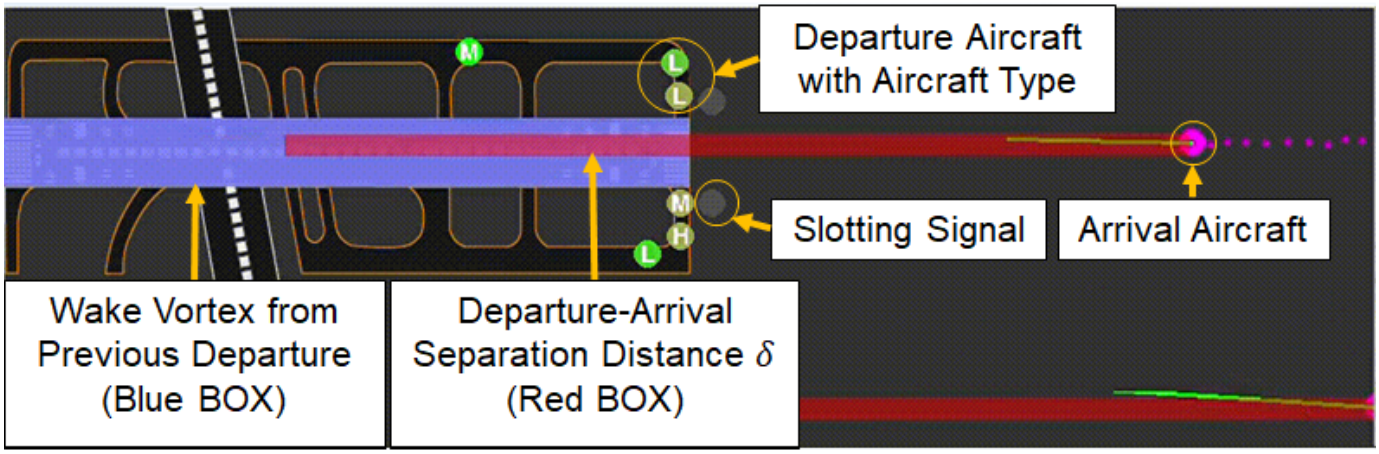


Figure 7: Encoding operational conditions into state representation by enhancing the image with various visualizations.

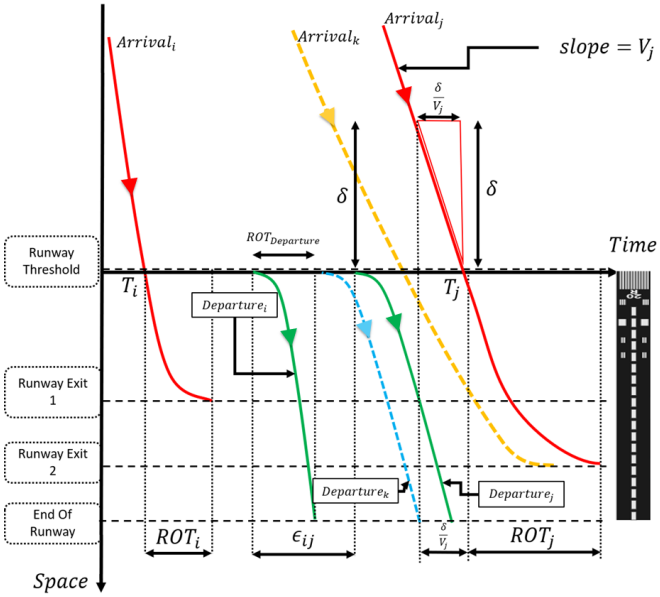


Figure 8: An illustration of a Time-Space Diagram for mix-mode runway operations.

design of the reward mechanism. The reward mechanism is designed as such to guide the model to achieve utilize the slots intelligently and optimally while eliminating any potential risks in violating the safety separation. Table IV provides detail information of the reward mechanism. The reward mechanism includes four components which are (1) the default each step small penalty, (2) separation-violation penalty, (3) slot assignment reward and (4) GCOL penalty. In which, the components (1) and (3) encourage the releasing decision. The second component, i.e.,violation penalty, shows the trade-off between quickly releasing a departure and violating safety separation minima. For instant, if the departure violates any separation constraints for more than 27 seconds ($\frac{2+2}{0.15} \approx 27$ time steps), the gain from successful slotting decision will be neglected. Finally, Ground Collision (GCOL) event is considered when two aircraft go too close to each other on the runway. It means the GCOL event only happens when the departure already violates the separation constraints for a certain amount of time. Therefore, that decision is considered as "not good" even without a GCOL event. Moreover, to support the training process at the early stage, namely "warm up", the GCOL penalty should not be too high. Since any extremely high penalty values will discourage the model to perform the exploration and exploitation, especially at its early stage when the naive model has a tendency to make several mistakes. In this work, the values in the final reward mechanism are adjusted experimentally while considering the mentioned relationships. As a result, during the training process, the model has learnt to optimize its reward by avoiding any separation violations and GCOL events.

TABLE IV: Reward Mechanism for Optimizing Departure Slotting

	Reward (r_t)
Ground Collision (GCOL)	-100
Arr-Dep Violation (per step)	-0.15
Dep-Arr Violation (per step)	-0.15
Dep-Dep Violation (per step)	-0.15
Step Penalty (per step)	$-(0.001 + 0.01 * N_{Lated\_Dep})$
Releasing Reward	2
Successful Departure	2
Timely Reward	$[-0.5, 0.5]$

V. Deep Reinforcement Learning Approach

The main goal of reinforcement learning is maximise the sum of future rewards (Equation 1) . The first step is to formulate the problem into a Markov Decision Process (MDP) and the main variables (State,Action,Reward,Next State) are returned at every timestep: (s, a, r, s') .

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

G_t = Discounted Sum of Future Rewards
γ = Discount Factor
r_t = Reward at timestep t

One popular family of reinforcement learning algorithms named Actor-Critic Methods uses the Value or Action-Value Function ($Q(s,a)$ or $V(s)$) as a critic for a policy ($\pi(a|s)$) update. A group of OpenAI researchers proposed the Proximal Policy Optimization (PPO) algorithm as an improvement to the Trust Region Policy Optimization [19] algorithm by introducing a new loss function named the "Clipped Surrogate Objective" that limits the policy update to prevent large unstable updates to the policy ($\pi(a|s)$) [20].The PPO paper used a Generalised Advantage Estimate [21] in place of Equation 4 which stabilize the Advantage Estimate (A) during training .

$$L^{CLIP}(\theta) = \mathbb{E}_t[min(R_\theta A, clip(R_\theta, 1-\epsilon, 1+\epsilon)A)] \quad (2)$$

where:

$$R_\theta = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} \quad (3)$$

$$A = r + \gamma V_\theta(s') - V_\theta(s) \quad (4)$$

If both the Policy $\pi_\theta(a|s)$ and Value Function $V_\theta(s)$ is parameterized by the same neural network with the weights θ, then the following combined loss function would be minimised during training:

$$L^{Total}(\theta) = -L^{CLIP}(\theta) + c_1 L^{VF}(\theta) - c_2(H(\pi_\theta)) \quad (5)$$

where $c1$ and $c2$ are parametric constants and $H(\pi_\theta)$ represents the entropy bonus to encourage exploration. L^{VF} represents the loss function of the Value function $V_\theta(s)$ and will be trained using a Squared Error Loss :

$$L^{VF}(\theta) = (r + \gamma V_\theta(s') - V_\theta(s))^2 \quad (6)$$

The Proximal Policy Optimization (PPO) deep reinforcement learning algorithm is used to train a model in the simulator environment described in Section IV. The convolutional neural network structure described in the Nature Deep-Q Learning paper [22] will be used as the initial layers of the Policy Network $\pi_\theta(a|s)$ and the Value Network $V_\theta(s)$. The various hyper-parameters will be listed in Section VI.

VI. Experimental Setting

The learning model is adopted from the stable-baselines reinforcement learning library [23] with the Tensorflow backend. The default PPO2 parameters in stable-baselines is used (Table V) and the neural network structure is illustrated in Figure 9.

Table VI shows the set of parameters which are used in the experiment. The Arrival Interval T is the time difference between generating two arrivals in the simulator and is chosen randomly from the range listed in this table. δ and ϵ are set with references from [18]. The required separation after

TABLE V: PPO2 Reinforcement Learning Parameters

PPO Parameters	Values
Training Iterations	1e+07
Number of Parallel Environments	20
γ	0.99
c1	0.5
c2	0.01
Learning Rate	0.00025
λ	0.95
ϵ	0.2
Optimizer	Adam

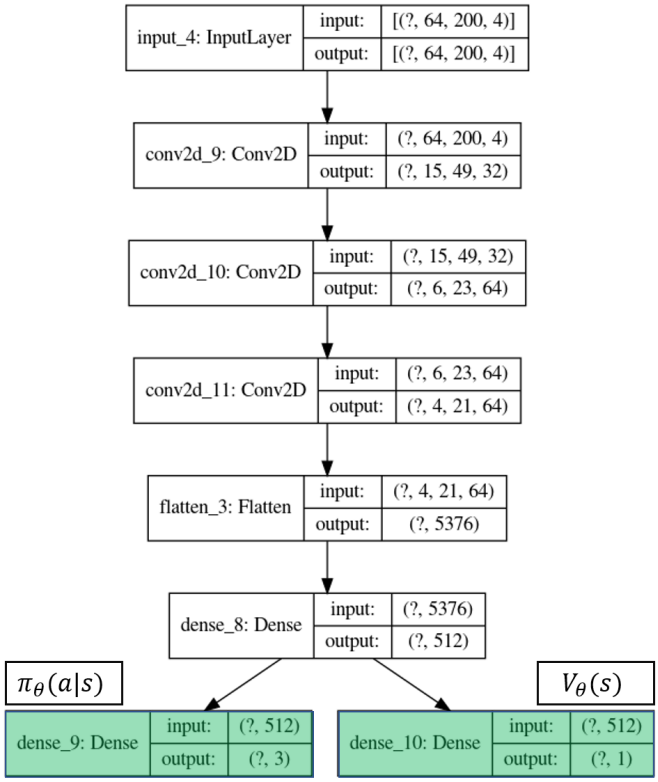


Figure 9: The illustration of applied network architecture for the learning model. The network includes seven layers with two output (action probabilities $\pi_\theta(a|s)$ and critic $V_\theta(s)$)

the heavy leading aircraft is 3 minutes while it is 2 minutes for medium and light aircraft. The mean $ROT_{departure}$ from A-SMGCS data is observed to be 110s and the departure trajectory is constant.

After the episode is started, at each time step, the model needs to identify the potential slot from the incoming stream of arrival. Based on the given state, the model can make slotting decision to release a waiting departure. Due to the random delay, that departure may wait up to 20 minutes before it starts to line-up. It will take 32 seconds for the aircraft to reach the runway before speed-up and take-off. The process will continue until the end of the episode. During evaluation, all important timestamps will be recorded for further analysis.

To access the efficiency of the trained model, the runway

TABLE VI: Values of Simulation Parameters.

Simulator Parameters	Values
δ	3NM
ϵ_{ij}	120s (i = H) and 60s (otherwises)
Arrival Interval T	$T \in \{130s, 131s, ..., 400s\}$
$ROT_{departure}$	$\mathcal{N}(110,\ 53)$
$N_{arrivals}$ (Per Episode Training)	20
$N_{arrivals}$ (Per Episode Testing)	500
Maximum Waiting Time	$T_w = 20$ minutes
Delays in Execution	$T_{DE} \in \{0s, 1s, ..., 20s\}$
Maximum Queue Length	$N_{MaxQueue} \in \{3, 4, 5, 6\}$
WTC Distribution	{ H: 0.33, M = 0.33, L = 0.33}
Queue Density (Departures/Hour)	$U(18, 36)$

capacity using the trained model is computed and compared to the estimated one from recorded episodes. Based on the probabilities of each WTC and the required separation ϵ_{ij}, the expected value of departure occupancy time $\mathbb{E}(TD)$ is $180s$. Let n_e^k is the number of expected departure slots of after an arrival (indexed k^{th} aircraft in the runway sequence, the total number of expected departure slots from the recorded data can be estimated as:

$$n_e = \sum_{k=1}^{N} n_e^k = \sum_{k=1}^{N} \frac{T_{ij}^k - \frac{\delta}{V_j^k} - ROT_i^{(k)}}{\mathbb{E}(TD)} \tag{7}$$

In which T_{ij}^k, $\frac{\delta}{V_j^k}$ and ROT_i^k are derived from recorded data for k^{th} arrival of the variables T_{ij}, $\frac{\delta}{V_j}$ and ROT_i (defined in TABLE III).

Let n_a denote the total arrivals and n_p denote the total predicted departure slots. Then the performance of the trained model is accessed based on the ratio R of predicted and expected runway capacities:

$$R = \frac{n_a + n_p}{n_a + n_e} \tag{8}$$

In addition, violations, in term of time separations when slotting departure flights of the trained model, could be observed by analysing the distribution of separations during the episode. This analysis will evaluate how the policy $\pi_\theta(a|s)$ balances between maximising safety and the runway usage. Let t_{Xi}^k be a random variable for the recorded timestamp of the k^{th} aircraft in the runway sequence when it reach the runway. In which $X \in \{D, A\}$ is the flight type and $i \in \{L, M, H\}$ is the WTC of that aircraft. The violations for Departure-Departure, Departure-Arrival and Arrival-Departure pairs are calculated as follows:

$$\Delta_{DD} = t_{Dj}^{k+1} - (t_{Di}^k + \epsilon_{ij}^k + ROT_{Departure}^k) \tag{9}$$

$$\Delta_{DA} = (t_{Aj}^{k+1} + \frac{\delta}{V_j^k}) - t_{Di}^k \tag{10}$$

$$\Delta_{AD} = t_{Dj}^{k+1} - (t_{Ai}^k + ROT_i^k) \tag{11}$$

In which ϵ_{ij}^k , $\frac{\delta}{V_j^k}$ and ROT_i^k are estimated values for corresponding arrival of ϵ_{ij}, $\frac{\delta}{V_j}$ and ROT_i (defined in TABLE III). Besides t_{Xi}^k, $\frac{\delta}{V_j^k}$ and ROT_i^k can be extracted directly from recorded data.

These Equations 9,10, 11 are defined such that any violation will lead to negative values. All non-negative values are corresponding to good decisions which satisfy the safety separation.

VII. Results and Discussions

The computational speed is more than 100 frames per second (Each frame in the simulator simulates one second of runway operation) on a standard Window laptop running on a Core i7 5500u during training and testing. It indicated that the trained model is not computationally heavy and would be suitable for real-time operations.

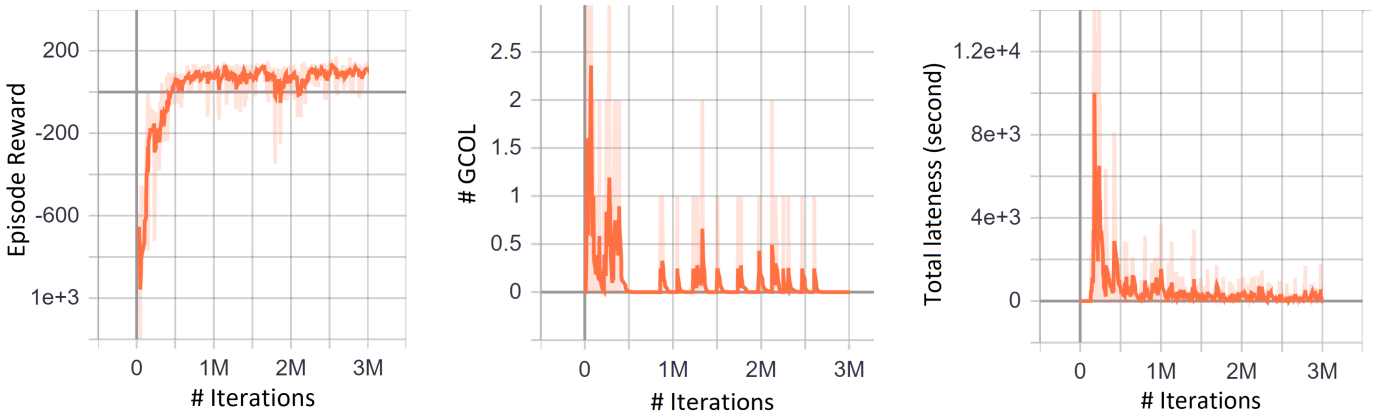
Figure 10 shows the convergence of the training process. After three millions iterations, the model has shown a stable performance in terms of episode reward, GCOL events and late departures. The model achieve zero GCOL events with high average episode score ($\approx$ 100) and the total extra waiting time (after reaching the maximum waiting time) for the episode is $\approx 150s$ on average. The model achieves Zero GCOL event for making more than 5000 departures. In which, the efficiency ratio R is greater than $83.8\% \pm 3.8$. This means that the trained model can achieve up to 83.8% of the expected (or theoretical) runway capacity given arrival sequences.

Table VII presents the basis statistics of the safety separation status from recorded data. There are three different cases are investigated: leading Departure - trailing Arrival, leading Arrival - trailing Departure and leading Departure - trailing Departure. The negative values (in seconds) mean the violations in term of safety separation standard. The distributions of those safety separation status are visualized in Figure 11.

Departure-Arrival Separation: There is no violation of the safety separation for Departure-Arrival Separation. It means the separation between two aircraft is always greater than $\sigma = 3NM$. Since the mean ROT of departures is 110s, the mean separation value (109s) implies that the arrival passes the runway threshold just at the time when the leading departure successfully takeoffs.

Arrival-Departure Separation: 0.18% of the trailing departures violates the safety separation with the maximum magnitude is 24 seconds. It means that the departure reaches the runway 24 seconds before the leading arrival exits the runway. Since the mean ROT of arrivals is 57s, the mean separation value (63s) implies that, on average, the trailing departure has waited until the arrival leaves the runway before speeding up.

Departure-Departure Separation: 0.11% of the trailing departures violates the safety separation with the maximum magnitude is only 5 seconds. It can be observed from the



(a) Convergence of Episode Reward (b) Convergence of GCOL event. (c) Convergence of late departures.

Figure 10: Illustrations for the convergence of the training process, including (a) Achieved Episode Reward, (b) Remaining GCOL Event, (c) Late Departures in Queues.

results that a buffer is usually added, automatically by the model, between two departures to account for departure ROT uncertainty. However, in case of too many late departures, slotting decisions can be made without considering the mentioned buffer.

TABLE VII: The summary of the learned model's separation performance

	Dep-Arr (s)	Arr-Dep (s)	Dep-Dep (s)
mean	109.39	62.62	40.97
std	58.28	60.39	6.17
min	2.00	-24.00	-5.00
25%	62.00	27.00	40.00
50%	104.00	41.00	41.00
75%	136.00	73.00	43.00
max	290.00	377.00	90.00

VIII. Conclusion

This research propose a Reinforcement Learning approach for real-time departure slotting in Mixed-Mode Runway Operation. A data-driven simulator , which is used for training and testing the model, showcases a novel state representation that graphically illustrates surveillance data (from A-SMGCS) and live operational conditions. The mode can make the slotting decision in multiple departure queues situations while also considering uncertainties from arrivals, departures and departure waiting time. A case study for Zurich Airport Runway 28 is developed and investigated. The results show that the trained model is able to achieve the efficiency ratio over 83.8% of the expected departure slots while maintaining a conservative policy that prefers safety over performance.

Acknowledgement

This research is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore.

References

[1] L. Adacher, M. Flamini, and E. Romano, "Airport ground movement problem: minimization of delay and pollution emission," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3830–3839, 2018.

[2] K. G. Zografos, M. A. Madas, and K. N. Androutsopoulos, "Increasing airport capacity utilisation through optimum slot scheduling: review of current developments and identification of future needs," *Journal of Scheduling*, vol. 20, pp. 3–24, 2017.

[3] ICAO. (2004) Manual on simultaneous operations on parallel or near-parallel instrument runways. https://www.icao.int/Meetings/AMC/MA/EleventhAir\%20Navigation\%20Conference\%20\%28ANConf11\%29/anconf11_ip003_app_en.pdf.

[4] G. F. Newell, "Airport capacity and delays," *Transportation Science*, vol. 13, no. 3, pp. 201–241, 1979.

[5] J. A. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson, "On-line decision support for take-off runway scheduling with uncertain taxi times at london heathrow airport," *Journal of Scheduling*, vol. 11, no. 5, p. 323, 2008.

[6] H. A. Limited. (2020) Return to single runway operations. [Online]. Available: https://www.heathrow.com/company/local-community/noise/latest-local-community-and-noise-news/return-to-single-runway-operations

[7] J. A. Bennell, M. Mesgarpour, and C. N. Potts, "Airport runway scheduling," *4OR*, vol. 9, no. 2, p. 115, 2011.

[8] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling aircraft landings—the static case," *Transportation science*, vol. 34, no. 2, pp. 180–197, 2000.

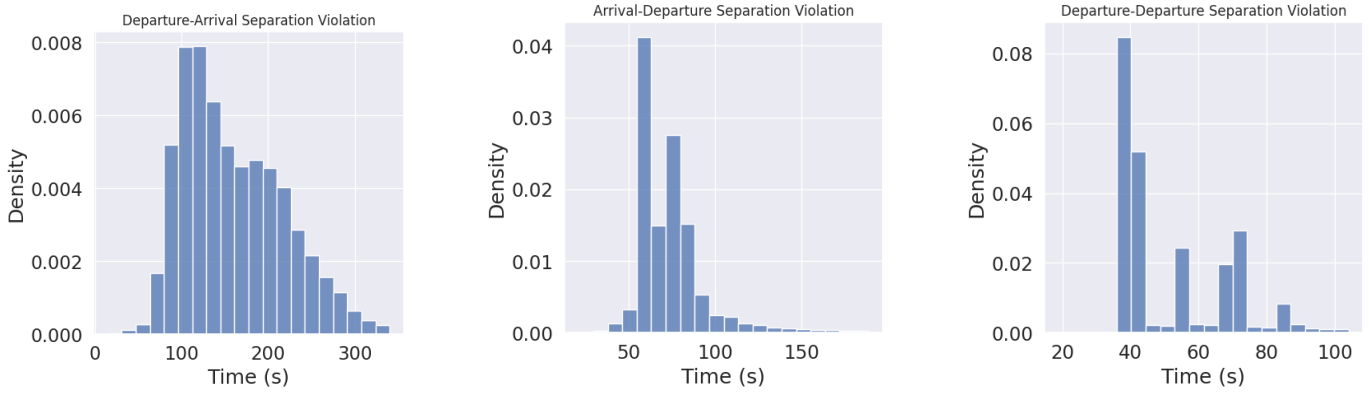
[9] M. Mesgarpour, C. N. Potts, and J. A. Bennell, "Models for aircraft landing optimization," in *Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT 2010)*, 2010.

[10] H. Balakrishnan and B. Chandran, "Scheduling aircraft landings under constrained position shifting," in *AIAA guidance, navigation, and control conference and exhibit*, 2006, p. 6320.

[11] H. Lee and H. Balakrishnan, "Fuel cost, delay and throughput tradeoffs in runway scheduling," in *2008 American Control Conference*. IEEE, 2008, pp. 2449–2454.

[12] J. E. Beasley, J. Sonander, and P. Havelock, "Scheduling aircraft landings at london heathrow using a population heuristic," *Journal of the operational Research Society*, vol. 52, no. 5, pp. 483–493, 2001.

[13] H. Helmke, O. Gluchshenko, A. Martin, A. Peter, S. Pokutta, and U. Siebert, "Optimal mixed-mode runway scheduling—mixed-integer programming for atc scheduling," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. IEEE, 2011, pp. 2C4–1.



(a) Departure-Arrival Safety Distribution (b) Arrival-Departure Safety Distribution (c) Departure-Departure Safety Distribution

Figure 11: Distributions of different separations for analysing the safety of proposed model.

[14] J. Djokic, B. Lorenz, and H. Fricke, "Air traffic control complexity as workload driver," *Transportation research part C: emerging technologies*, vol. 18, no. 6, pp. 930–936, 2010.

[15] P. Balakrishna, R. Ganesan, L. Sherry, and B. S. Levy, "Estimating taxi-out times with a reinforcement learning algorithm," in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. IEEE, 2008, pp. 3–D.

[16] M. Bloem and N. Bambos, "Ground delay program analytics with behavioral cloning and inverse reinforcement learning," *Journal of Aerospace Information Systems*, vol. 12, no. 3, pp. 299–313, 2015.

[17] S.-L. A. Tien, H. Tang, D. Kirk, E. Vargo, and S. Liu, "Deep reinforcement learning applied to airport surface movement planning," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, 2019, pp. 1–8.

[18] A. A. Trani, *Airport Capacity - Introduction*, 2012, http://128.173.204.63/courses/cee5614/cee5614_pub/Airport_capacity_intro_2012.pdf.

[19] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015, pp. 1889–1897.

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

[21] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[23] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.

AUTHOR BIOGRAPHY

Duc-Thinh Pham obtained his Ph.D. degree of University Paris Science Lettres (PSL) in computer science, France, in 2019. He is currently Research Fellow at Air Traffic Management Research Institute (ATMRI), Nanyang Technological University (NTU). His research interest are applications of machine learning in air traffic management such as surface movement management and conflict resolution. He is the Assistant Program Director of AI & DA for ATM Program.

Li Long Chan graduated B.Eng Aerospace Engineering in 2020 from the School of Mechanical and Aerospace Engineering (MAE), NTU. He is currently Research Assistant at ATMRI, NTU. His research focuses on machine learning, deep reinforcement learning and their applications in the field of Air Traffic Management, especially, Airport-Airside Management.

Sameer Alam is Associate Professor at the MAE, NTU Singapore. He received Ph.D. degree (2008) in Computer Science with specialization in Artificial Intelligence from University of New South Wales (UNSW) Australia. He is the Deputy Director of ATMRI, NTU Singapore. He is also the editorial board member of Transportation Research Part-C. His research interests are in Machine Learning, Computer Vision, Multi-Agent Systems, applied to Air Traffic and Airport operations.

Rainer Koelle is Head of Section Operational ANS Performance (Safety and Quality of Service) at EUROCONTROL Performance Review Unit. He received Ph.D. degree (2012) from Lancaster University, UK. He joined EUROCONTROL, Brussels, in 2005 as a ATM Security Expert and transferred to Performance Review Unit in 2012. His research focus is on time-critical decision making in aviation security, decision making under uncertainty and real-time threat assessment.