Taxiing Speed Intelligent Management of Aircraft Based on DQN for A-SMGCS

Yunfan Zhou, Weilong Liu, Yandong Li and Bo Jiang*

College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan, 618307, China
610578962@qq.com

Abstract. In view of ever-increasing air traffic, much attention has been paid to air traffic management research to improve efficiency. Normally, once two taxiing aircrafts may conflict in the hotspot area, one of them must wait in the holding point until the conflict is resolved. However, this method reduces the efficiency though it ensures sufficient safe distance between aircrafts. In this paper, we discussed the feasibility of the dynamic control of the taxiing speed to improve the efficiency under the A-SMGCS (Advanced Surface Movement Guidance and Control Systems). The conflict samples were trained based on actual operation data and AirTOp simulation operation data. Action spaces, state spaces and reward function were designed and DQN (Deep Q Network) model was used to make speed regulation decisions. The experimental results show that the model can adopt different strategies in different environments, which greatly improved the intelligent decision-making level of aircraft taxiing speed adjustment and the operating efficiency of the scene. The case study showed that the time of conflict solving and the main taxiway occupation length can be reduced by 13.16% and 47.9% separately when the conflict is released.

1. Introduction
As the rapid development of air transportation, the taxiway structure is becoming more and more complex and the number of intersections is also increasing. ICAO defines airport hotspot as the area where has more potential risk of collision or invasion of runway in the region of ground of the airport operation so far in running history. The intermediate holding position which used for ensure the safety margin can improve the safe operation level. However, it has a negative impact on the operating efficiency. As the taxiing aircraft has the characteristic of "no overtaking in one direction", once the aircraft waits standby during the taxiing, it will congest the next series of taxiing aircrafts.

In order to improve the operational efficiency of the airport and ensure the safety level at the same time, ICAO proposed Advanced Surface Movement Guidance and Control Systems (A-SMGCS). A-SMGCS[1] which includes functions of surveillance, control, routing and guidance, represents the most advanced concept of scene operation monitoring. A-SMGCS contains high precision monitoring module which can realize real-time monitoring, it formed a visual ground target activity situation based on the data fusion[2] from the SMR (Scene Surveillance Radar), MSSR (Monopulse Secondary Surveillance Radar), ADS-B (Automatic Dependent Surveillance-Broadcast) e.t. The positioning accuracy of A-SMGCS can reach the level of meter, and the information can be shared and processed between traffic and users for coordination and communication[3]. A-SMGCS can obtain reliable data from various management support systems and effectively integrate automatic information in airport,
so it can realize trajectory prediction[4] by means of multi-module integration advantages such as monitoring, control and routing plan[5].

Self-driving vehicle control has made dramatic progress in the last several years, the three-tier architecture which includes Strategic level, Tactical level and Operational level is commonly used in driver behavior research to describe the algorithm system. Tactical level has three parts: Perception, Decision and Control. If we want to introduce autopilot technology into the ground taxiing of aircraft, A-SMGCS can achieve the “perception”, Machine Learning must be used to realize “Decision”. Machine Learning such as reinforcement learning(RL) and deep Q network(DQN) has been demonstrated to be an effective way to optimize the management of ground traffic in several previous works such as speed control[6], automated lane change decision making[7], path planning[8], and cooperative intelligent freeway traffic control[9], etc. RL also has achieved success in controlling robotic motion[10] and playing games such as Go[11, 12] and Atari[13, 14].

In this paper, the DQN is proposed to enable the speed control of aircraft ground taxiing process. When the two aircrafts converge toward the conflict hotspot, the DQN can adjust the taxiing speed of the aircraft to reduce the frequency of aircraft waiting in the intermediate holding position, it can improve the efficiency under the premise of ensuring the safety margin.

2. Q Learning and Deep Q Network Model

Q learning[15] is a form of model-free reinforcement learning. It can also be viewed as a method of asynchronous dynamic programming (DP). The research object of reinforcement learning is sequential decision problem. As is shown in Figure 1, Q learning contains three elements: State, Action and Reward. Learning proceed means an agent tries an action at a particular state, evaluates its consequences in terms of the immediate reward or penalty and its estimate of the value of the state to which it is taken. Agent is always in a certain state (s) at a certain moment, and has a series of actions(a). The system has a reward r(s, a) for each state and action, each training will be updated iteratively. Finally, the state action evaluation function Q(s, a) is obtained. Establish the value table of "State→Action", namely the Q table, which will be close to the real Q'(s, a) after convergence.

![Figure 1. The process of reinforcement learning](image)

The applicability of Q learning has been limited to domains in which useful features can be handcrafted, or to domains with fully observed, low-dimensional state spaces because Q learning is based on discrete actions and state space. In the training process, all possible actions of each state need to be evaluated. If the state space is too large, especially in the case of continuous space or action, the results will be instability or even unsolvable. In order to realize the dynamic control of speed, the concept of DQN must be introduced. Volodymyr Mnih[16] developed a novel agent, a deep Q-network (DQN), which can combine reinforcement learning with a class of artificial neural network known as deep neural networks. DQN[17, 18] uses neural network to describe State value instead of Q table to find the value of "State→Action" by introducing “Experience Replay”[19] and target network.

Deep learning requires a large number of samples, and the introduction of “Experience Replay” can increase the sample size. The function forms a training sample e = (s, a, r, s'), which consists of the state, action, reward and the next state at each time point. Assuming that the model has performed N steps in total, a time playback sequence E is formed E = {e_1, e_2, e_3, ..., e_N}. During training, small
batch of experience samples are extracted from the sequence for network to learn. The sample is completed by setting up a queue in the form of buffer. The sample is pushed on the stack in a “first-in, first-out” manner, which can ensure that the training is to learn the previous experience comprehensively and repeatedly, this manner can also avoid learning only from the most recent sample.

DQN has two networks with the same structure [20]: target network and Q network. Both of them are deep Convolutional Neural Network(CNN). Nevertheless, only the Q network has the training process while the target network just simply copy the state of Q network. The parameters of the target network \( \theta_f \) are updated with the parameters of Q network \( \theta_i \) every N iterations, while the parameters of the intermediate process \( \theta_i \) remain unchanged. Where \( \theta_i \) is the parameter after the i iteration of Q Network, and \( \theta_f \) is the parameter of the target network.

\[ Q \text{ learning uses a max operator to select which action results in the largest potential future reward.} \]

Let \( x_i \) represent the observation results after continuous t-step of the model, then the components of Q learning are as follow:

\( s_i - a_i \) represents the action performed after \( x_i \) is observed at time t

\( A - \mathcal{A} \) is a set of all possible action, \( a_i \in \mathcal{A} \).

\( r_i - r_i \) represents the reward after \( x_i \) is observed at time t

\( R_i - R_i \) represents the accumulated value from the beginning time t

\[ R_i = \sum_{t=1}^{i} \gamma_i r_i \]  

\( \gamma_i \in [0,1] \), \( \gamma_i \) is discount factor determines how far in the future to look for rewards. When \( \gamma = 0 \), only immediate rewards are considered, whereas, when \( \gamma \to 1 \), future rewards are getting prioritized.

\( s_i - s_i \) represents the state at time t

\[ s_i = (x_i, a_1, ..., x_{t-1}, a_{t-1}, x_t) \]  

\( Q(s_i, a_i) - Q(s_i, a_i) \) is function determines State->Action

The essence of Q learning is the process of solving \( Q(s_t, a_t) \), we use the following formula to learn State->Action recursively:

\[ Q_{next} = (1 - \sigma) \cdot Q_{now}(s_t, a_t) + \sigma \cdot \delta \]

\[ \delta = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_t) \]  

In the formula (3), \( \sigma \) is the learning rate, the smaller \( \sigma \) is, the smaller the proportion of historical value is. \( \delta \) represents the learning target. \( s_t \) and \( a_t \) represent the state and action to be executed after the t-step iteration respectively.

3. Intelligent Speed Regulation Strategy Model

3.1. Purpose of model and parameter settings

In this research, we use the DQN model to resolve the conflict in hotspot as shown in Fig. 2. In the current conflict resolution processing, the priority of the straight-ahead aircraft is generally guaranteed, and the turning aircraft will wait in the intermediate holding position until the aircraft straight-going has passed the intersection. Since the run-up time of the aircraft is much longer than the acceleration time when the aircraft is taxiing with low speed. When the two aircrafts converge toward the conflict hotspot O, the agent adjusts the taxiing speed of the aircraft to reduce the frequency of aircraft waiting in the intermediate holding position.

Aerodrome Technical Standards[21] specifies, when the aircraft movement area reference code II is F, and when traffic density is high, clearance between taxiway and the runway, taxiway and other objects should not be less than the specified value in the table 1. When determining the intermediate holding position, it shall be ensured that the clearance between the aircraft taxiing on the intersecting taxiway must meet the requirements of the standards[22]. When conflicts cannot be avoided, the aircraft must stop in time before the intermediate holding position. The model of BADA is adopted to
calculate the ground operation model of the aircraft. Taking B-738 as an example, the specific data of this model are as follows.

![Figure 2. Diagram of conflict](image)

| Parameter Name                                      | Parameter Values |
|-----------------------------------------------------|------------------|
| The clearance between the center line of a taxiway and another taxiway | 97.5m           |
| The clearance between the center line of a taxiway and an object     | 57.5m           |
| Maximum taxiing speed                                             | 50km/h(13.89m/s) |
| Maximum turning speed                                              | 20km/h(5.56m/s)  |
| Environmental renewal cycle                                       | 1s              |
| Windspan of aircraft                                               | 34.10m          |
| Length of the fuselage                                             | 11.76m          |
| Radius of turning circle                                           | 22.9m           |
| Acceleration                                                       | 1m/s            |

3.2. Description of Reinforcement Learning

3.2.1 State Space. The description of state space adopts Cartesian coordinate system which is based on bird’s-eye-view, and each state S is composed of six components: taxiing speed of aircraft A (straight going), horizontal coordinate of aircraft A, vertical coordinate of aircraft A, taxiing speed of aircraft A (turning), horizontal coordinate of aircraft B, vertical coordinate of aircraft B.

\[ S = \{ V_A, x_A, y_A, V_B, x_B, y_B \} \] (4)

3.2.2. Action Space. The description of action space is described by three components: \{acceleration, uniform speed, deceleration\}. This model is simplified by set the speed of acceleration to \(1m/s^2\).

\[ A = \{ a^+, a, a^- \} \] (5)

3.2.3. Reward Function. Considering the available time of conflict resolution and the degree of conflict risk, the design formula of the reward function is as follows:

\[ r_i = (1-\sigma) \cdot T(s,r) + \sigma \cdot d(s,r) \] (6)

\[ T(s,r) = \min ((x_{\text{crossing}} - x_{A,s}) \left( \frac{\partial (d_{A,s} - d_{\text{min}})}{\partial x} \right)^{-1}, (y_{\text{crossing}} - y_{B,s}) \left( \frac{\partial (d_{B,s} - d_{\text{min}})}{\partial y} \right)^{-1}) \] (7)

\[ d(s,r) = \frac{d_x \cdot d_{\text{min}}}{d^2_x} \] (8)
In (7), \(T(s, r)\) is the reward function measuring the time available for releasing conflict under the current state, \(d(s, r)\) is the reward function measuring the distance available for releasing the conflict, where the distance between two aircraft is defined as: 
\[d = \left( (x_A - y_A)^2 + (x_B - y_B)^2 \right)^{1/2},\]
and \(d_a\) represents the braking distance required for deceleration at the maximum deceleration speed.

The setting of additional rewards function is divided into reward and punishment. In this model, the reward is set to taxi without waiting till passing through the hotspot, and the penalty is set to the distance lesser than the safety threshold. If both aircrafts pass through the intersection point safely and keep the safe clearance, an additional bonus value of +10 will be given to the Agent [23]. During the experiment (as shown in Figure 2), if the spacing between the two aircraft is less than 100m before passing the intersection, it will be judged as dangerous proximity. The model will give an alarm and end the experiment, and the Agent will be given an additional penalty value of -10.

### 3.3. Empirical Evaluations

In the deep Q network, Q network and target network should be constructed. The Q network is updated every episode, and the target network is updated occasionally. Deep Q network operation flow is shown in Figure 3 and parameters of neural network are shown in table 2.

![Deep Q network operation flow chart](image)

**Figure 3. Deep Q network operation flow chart**

| Parameter Name                                                                 | Parameter Values           |
|-------------------------------------------------------------------------------|---------------------------|
| Range of distance between the initial position aircraft straight going and intersection | 200m-400m                 |
| Range of distance between the initial position aircraft turning and intersection | 75m-250m                  |
| Neural network framework                                                      | TensorFlow                |
| Learning rate \(\sigma\)                                                      | 0.01                      |
| Discount factor \(\gamma\)                                                    | 0.9                       |
| The probability of greedy policy \(\varepsilon\)                              | 0.9                       |
| replay memory size                                                            | 2000                      |

The replay memory size of the architecture was 2,000 with a replay start size of 200 random actions and a mini-batch size of 20. The default value for the target network update frequency was 1200; the replay memory is used for independent processing of data, to weaken the strong dependence between adjacent episodes and increase the learning efficiency.

A neural network model with three fully-connected layers was constructed. The input layer consists of xx neurons to combine the position and velocity parameters of the aircraft. The number of neurons in the hidden layer was set to 9 to obtain the convergence of the whole network. The output layer with 3 neurons is used to provide the predicted actions. All the agents behaved and estimated the action.
values with an $\varepsilon$-greedy policy where $\varepsilon = 0.9$. And we set a discount factor $\gamma$ of 0.9 to encourage agents to find the terminal state.

4. Model application and analysis of experimental results

4.1. The Data Source of the Model
The model which is verified in this paper is based on the hotspots deprived from single runway operation, independent parallel operation, segregated parallel operations and dependent parallel operation in an airport in the Central and South part of China. The AirTOp simulation software which is widely used in the civil aviation industry is utilized to conduct hotspots of conflict in the simulation operation. Considering the data of the survey and software parameters to analyzes the details of the holding, the conflicts on the main taxiway when the aircraft turning from the aircraft position are put at the core.

4.2. The Analysis of the Result
Figure 5 shows that as episodes increase, the rising rate in the number of successes is significantly faster than the rising rate of failure cases. In addition, it shows in Figure 6 that the success rate is larger in the first 4000 episodes, and then the growth begins to slow down. The plausible reason is that when the number of rounds is rather smaller, the position of the initial aircraft has a greater impact on the training results, followed by the depth Q network model. Furthermore, the learning ability of DQN is not strong enough when the number of rounds is small, and after 6,000 episodes, the success rate can be maintained, fluctuating around about 67%.
This section of the experiment mainly verifies the strategy of the trained model when dealing with different speeds. In the initial state of the environment, the initial speed of the taxiing aircraft is 50km/h (13.89m/s), the aircraft A (straight going) is 250m away from the intersection, and the aircraft turning (B) is 150m away from the intersection. During the operation of the AirTOp simulation, the aircraft B decelerates to the speed of 0 (shown as the I area of the Figure) until it stops before the intermediate holding position (shown as the area II). After the aircraft A passes the intersection, the aircraft B starts and turns through the intersection (shown as the area III). It takes 38s to release the conflict. When the conflict is released, the occupancy length of the main taxiway is 188.7m.

In the deep Q network optimization scheme, the aircraft A is still allowed to taxi at its maximum speed. The aircraft B first decelerates and then taxis at a low speed. After the aircraft A passed through the intersection, it continues to decelerate to the maximum turning speed without holding, and the conflict is fully released in 33s. After the conflict is released, the occupancy length of the main taxiway is 98.4m. The time consumed in this scheme is 13.16% lesser if compared with the simulation data, and the occupancy length of the main taxiway can be reduced by 47.9% after the conflict is released.

5. Conclusion
In this paper, the Deep Q Network model is used for speed control of aircraft ground taxiing process. By training the DQN with the conflict samples in the ground operation, and designing the action space, state space and reward function, the experimental results show that the DQN can adopt different strategies for the aircraft when facing different environments. Under the premise of ensuring safety margin, the model can reduce the waiting time of the ground by about 60%. Compared with the AirTOp software simulation operation data, the time used both in the process of the conflict resolution and in the occupancy length of the main taxiway decline significantly. Although this paper only exposes control over the speed of two aircrafts in conflicts, the operation of multiple aircrafts is actually a much more challenging for the structure of this neural network. However, the utility of using this solution for speed control of multiple aircraft needs further exploration. In addition, whether the speed regulation will increase the workload of controllers and the pressure of monitoring or not still needs further research.

Acknowledgments
Joint Funds of the National Science Foundation of China and the Civil Aviation Administration (Grant No. U1733105); National Innovation and Entrepreneurship Training Program for College Students of Civil Aviation Flight University of China (Grant Nos. S201910624067, 201810624001); Scientific General Program Foundation of Civil Aviation Flight University of China (Grant No. J2019-009)

References
[1] Jakobi, J., Teotino, D., Montebello, P. Higher-level services of an advanced surface movement guidance and control system (A-SMGCS). Air Traffic Control Quarterly. vol. 152, no.
Li, J., Zhao, X., Wu, M., Yu, E., Jiang, L., Sheng, P. (2017). Design and Implementation of A-SMGCS Based on ICE. In: MATEC Web of Conferences. EDP Sciences. Vol. 108, pp. 10004

Pan, W.J., Wang, X., Luo, X.L. (2016) Taxiing Route Planning Based on Airport Hotspots. In: 30th Congress of the International Council of the Aeronautical Sciences. Daejeon, pp. 1-4

Okuniek, J. N., Gerdes, I., Jakobi, J., Ludwig, T., Hoey, B. L., Foyle, D., Zhu, Z. (2016). A concept of operations for trajectory-based taxi operations. In: 16th AIAA Aviation Technology, Integration, and Operations Conference, pp. 3753.

Young, S. D., Jones, D. R. (2000). Runway incursion prevention using an advanced surface movement guidance and control system (A-SMGCS). In: 19th DASC. 19th Digital Avionics Systems Conference. Proceedings Vol. 1, pp. 2E2-1.

Chen, P., He, Z., Chen, C., Xu, J. (2018). Control Strategy of Speed Servo Systems Based on Deep Reinforcement Learning. Algorithms, 11(5), pp. 65.

Hoel, C. J., Wolff, K., Laine, L. (2018). Automated speed and lane change decision making using deep reinforcement learning. In: 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE. pp. 2148-2155.

Lv, L., Zhang, S., Ding, D., Wang, Y. (2019). Path Planning via an Improved DQN-based Learning Policy. IEEE Access.

Wu, Y., Tan, H., Jiang, Z., Ran, B. (2019). ES-CTC: A Deep Neuroevolution Model for Cooperative Intelligent Freeway Traffic Control. arXiv preprint arXiv:1905.04083.

Zhang, F., Leitner, J., Milford, M., Upcroft, B., Corke, P. (2015). Towards vision-based deep reinforcement learning for robotic motion control. arXiv preprint arXiv:1511.03791.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), pp. 484.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Chen, Y. (2017). Mastering the game of Go without human knowledge. Nature, 550(7676), pp. 354.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), pp. 529.

Hoos, I. A., Rebedea, T. (2016). Playing atari games with deep reinforcement learning and human checkpoint replay. arXiv preprint arXiv:1607.05077.

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Baomar, H., Bentley, P. J. (2016). An Intelligent Autopilot System that learns flight emergency procedures by imitating human pilots. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE. pp. 1-9.

Nishio, D., Yamane, S. (2018). Faster deep q-learning using neural episodic control. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). Vol. 1, pp. 486-491.

Hernandez-Garcia, J. F., Sutton, R. S. (2019). Understanding Multi-Step Deep Reinforcement Learning: A Systematic Study of the DQN Target. arXiv preprint arXiv:1901.07510.

Lin, L. J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. Machine learning, 8(3-4), 293-321.

Brittain, M., Wei, P. (2018). Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning. In: Proceedings of the International Conference for Research in Air Transportation.

Civil Aviation Administration of China. (2013) Aerodrome technical standards

International Civil Aviation Organization. (2016) Air Traffic Management

Sallab, A. E., Abdou, M., Perot, E., Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19), pp. 70-76.