Improving Autonomous Separation Assurance through Distributed Reinforcement Learning with Attention Networks

Marc W. Brittain, Luis E. Alvarez, Kara Breeden

Massachusetts Institute of Technology Lincoln Laboratory*
marc.brittain@ll.mit.edu, luis.alvarez@ll.mit.edu, kara.breeden@ll.mit.edu

Abstract

Advanced Air Mobility (AAM) introduces a new, efficient mode of transportation with the use of vehicle autonomy and electrified aircraft to provide increasingly autonomous transportation between previously underserved markets. Safe and efficient navigation of low altitude aircraft through highly dense environments requires the integration of a multitude of complex observations, such as surveillance, knowledge of vehicle dynamics, and weather. The processing and reasoning on these observations pose challenges due to the various sources of uncertainty in the information while ensuring cooperation with a variable number of aircraft in the airspace. These challenges coupled with the requirement to make safety-critical decisions in real-time rule out the use of conventional separation assurance techniques. We present a decentralized reinforcement learning framework to provide autonomous self-separation capabilities within AAM corridors with the use of speed and vertical maneuvers. The problem is formulated as a Markov Decision Process and solved by developing a novel extension to the sample-efficient, off-policy soft actor-critic (SAC) algorithm. We introduce the use of attention networks for variable-length observation processing and a distributed computing architecture to achieve high training sample throughput as compared to existing approaches. A comprehensive numerical study shows that the proposed framework can ensure safe and efficient separation of aircraft in high density, dynamic environments with various sources of uncertainty.

*Distribution Statement A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force. Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

Introduction

Advanced air mobility (AAM) is set to revolutionize transportation by introducing highly automated aircraft to transport passengers and cargo within local, regional, interregional, and urban environments (Federal Aviation Administration 2023). However, the realization of AAM faces several key challenges, including safety, security, social acceptance, resilience, environmental impacts, regulation, scalability, and flexibility (National Academies of Sciences, Engineering, and Medicine 2020).

To address these challenges, the use of advanced automation techniques such as artificial intelligence (AI) is essential. Specifically, learning-based decentralized separation assurance holds significant potential for enabling the safe and efficient operation of highly automated aircraft in the high-density, high-tempo AAM environment envisioned by the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA) (Federal Aviation Administration 2023; National Aeronautics and Space Administration 2020).

However, due to the lack of real operational data and scenarios for AAM, developing, training, and validating AI algorithms becomes a challenge. Simulation provides a low-cost way to overcome this challenge by allowing for the exploration of edge-case scenarios that may be too dangerous to perform in the real world. Therefore, simulation-based training and validation of AI algorithms are essential for safe and efficient operation of AAM. Recently, the AI testbed for Advanced Air Mobility (AAM-Gym) was developed to provide a standardized ecosystem for the research of AI in AAM. By leveraging simulation backends such as BlueSky (Hoekstra and Ellerbroek 2016) and UAM-Toolkit (Alvarez et al. 2021), representative real-world scenarios can be developed for training and evaluation.

Recently, deep reinforcement learning (DRL) has demonstrated superior performance to humans in games such as Atari, GO, Warcraft, and StarCraft II, requiring a sophisticated balance between near-term and long-term strategic decisions (Mnih et al. 2013; Silver et al. 2016; Amato and Shani 2010; Vinyals et al. 2017). In addition, DRL has also been applied to air traffic control (ATC) and conflict reso-

lution, where early work used an AI agent to mitigate conflicts and minimize the delay of aircraft reaching their metering fixes (Brittain and Wei 2018). In later works, (Pham et al. 2019) demonstrated that an AI agent can effectively resolve randomly generated conflict scenarios between a pair of aircraft through vectoring maneuvers. To encourage human ATC adoption of AI maneuvers, (Tran et al. 2020) developed an interactive conflict solver using DRL that was trained using human resolution maneuvers, providing AI behavior more closely aligned with humans. More recently, (Ribeiro, Ellerbroek, and Hoekstra 2020, 2022) proposed a hybrid geometric-reinforcement learning algorithm for resolving conflicts in low-altitude airspace. (Badea et al. 2022) explored the use of both lateral and vertical maneuvers for conflict resolution using DRL in traditional airspace. While these approaches are effective for sparse airspace environments, they fail to handle state space scalability as the number of intruder aircraft increases due to the either centralized, single-agent architectures or fixed-length state vectors with a maximum number of intruder aircraft. In (Brittain and Wei 2019; Brittain and Wei 2021; Brittain, Yang, and Wei 2021; Brittain and Wei 2022), it is shown how a decentralized separation assurance framework can alleviate the aforementioned scalability concerns and prevent loss of separation in high-density stochastic sectors by leveraging long short-term memory networks (LSTM) and attention networks, even when agents may be optimizing non-homogeneous reward functions.

In this article, a decentralized learning-based framework for aircraft separation assurance is introduced and applied to a high-density AAM use-case. We integrate the sample efficient Discrete Soft Actor-Critic (SACD) algorithm and extend the algorithm with the use of attention networks to handle a variable-length state space. Given SACD is an off-policy algorithm, an asynchronous training architecture is developed to decouple the agent-environment interaction with the algorithm training. This allows us to achieve an approximately 10x increase in the number of transitions trained over existing approaches. We show that the increased training leads to improved safety and operational suitability performance, even in highly uncertain environments. The main contributions of this article are summarized as follows:

- We propose a scalable, distributed, and sample efficient aircraft separation assurance framework based on SACD and attention networks that is capable of both improving safety and operational suitability.
- We introduce an expanded action set over prior works with the introduction of vertical maneuvers.
- A representative AAM environment is developed in AAM-Gym, providing a comprehensive environment for evaluating the effectiveness of the proposed framework.

The structure of this paper is as follows. We first provide a brief overview of reinforcement learning and soft actor-critic. Then, we introduce the approach to applying reinforcement learning to aircraft separation assurance. Following, details on the environment setup and numerical experiments are presented. We then discuss the results and summarize our findings in the conclusion.

Background

In this section, we briefly review the background of reinforcement learning and soft actor-critic.

Reinforcement Learning

Reinforcement learning (RL) is one type of sequential decision making where the objective is to learn a policy in a given environment. RL requires the environment to be formulated as a Markov Decision Process (MDP); a mathematical framework for modeling decision making processes with stochastic transitions. An MDP is defined by the tuple (S, A, R, T, γ), where an agent in state $s \in S$ takes an action $a \in A$, transitions to state s' with probability $T(s'|s, a)$, and receives a reward $R(s, a)$. In RL, the transition matrix T is often unknown. The discount factor γ determines how far in the future to look for rewards, where immediate rewards are emphasized as $\gamma \to 0$ and future rewards are prioritized when $\gamma \to 1$.

The RL agent is able to derive an optimal policy π^* in the environment by maximizing a cumulative reward function

$$\pi^* = \arg\max_{\pi} E[\sum_{t=0}^{\tau}(r(s_t, a_t)|\pi)], \tag{1}$$

where τ represents the total time for a given environment. In environments with discrete or low-dimensional state-action representations, the optimal policy π^* can be obtained using dynamic programming approaches such as Q-learning (Watkins and Dayan 1992). However, many real-world environments can not be represented by discrete values or require high-dimensional state representations, requiring the use of function approximation for the policy. The aforementioned issues can be addressed through deep reinforcement learning (DRL) where a neural network is used to represent the policy π.

Soft Actor-Critic

Soft actor-critic (SAC) is a state-of-the-art off-policy deep reinforcement learning algorithm that has shown promise across a wide variety of continuous control tasks (Haarnoja et al. 2018a,b) and recently discrete action settings (Christodoulou 2019). Unlike traditional reinforcement learning, SAC seeks to derive an optimal policy based on a maximum entropy objective function

$$\pi^* = \arg\max_{\pi} E[\sum_{t\geq 0} \gamma^t (r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t)))], \tag{2}$$

where $\mathcal{H}(\pi(\cdot|s_t))$ represents the entropy of the policy distribution for a given state s_t such that

$$\mathcal{H}(\pi(\cdot|s_t)) = E[-\log(\pi(\cdot|s_t))]. \tag{3}$$

The temperature parameter α represents the trade-off coefficient between expected returns and the entropy term. The standard RL objective is recovered when $\alpha \to 0$.

While SAC and SACD perform well in many environments, the performance is greatly sensitive to the choice of α and subsequently the target entropy. Recent work by (Xu et al. 2021) introduced a target entropy annealing approach

to address the sensitivity of the target entropy parameter. In this work, we adopt the use of target entropy annealing, which we found to be essential to obtain good performance in the air transportation environment.

Approach

In this section, the aircraft separation assurance problem is introduced and formulated as an MDP by defining the state space, the action space, and the reward function. We then detail the distributed asynchronous training setup used to achieve a 10x increase in training throughput.

Aircraft Separation Assurance

Separation assurance involves preventing a loss of separation (LOS) event with aircraft in trail, at intersections, and at metering fixes by providing advisory maneuvers to aircraft. Any given aircraft in the environment is referred to as an ownship with all other aircraft in the airspace referred to as intruder aircraft from the ownship's point of view. In this way, each ownship will have its own associated intruder aircraft and selected action. The LOS threshold, d^{LOS}, defines a safety radius around ownship where operations with an intruder within the threshold become increasingly dangerous. Violating the loss of separation threshold may result in collisions between aircraft or near midair collisions (NMACs) that often result in drastic maneuvers from the aircraft. This task is traditionally performed by human air traffic controllers; however, novel automation techniques are required to safely scale to the expected magnitude of air traffic for AAM.

State Space

In order to provide a scalable solution for increasing air traffic, we adopt a centralized training, decentralized execution scheme where each aircraft is considered an agent and with training, learns a cooperative policy for navigating through the airspace safely and efficiently. The state space for this environment consists of ownship information as well as information from the surrounding air traffic that is dynamic in size as aircraft take-off and land. The state is therefore decomposed into the ownship state and the intruder state. To handle variability in the intruder aircraft information, we adopt the use of attention networks (Luong, Pham, and Manning 2015), similarly to the D2MAV-A algorithm (Brittain, Yang, and Wei 2021). This resulting intruder attention vector provides a fixed-length vector representation for network optimization. The ownship state space s at time t is defined as

$$s_t = (\psi, z, \dot{v}_x, \dot{v}_z, v_x, v_z, gs_{\text{East}}, gs_{\text{North}}, t, a_{t-1}, x_{\text{wpt}}(j) - x, y_{\text{wpt}}(j) - y) \quad \forall\, j \in [1, N_{\text{wpt}}],$$

to include heading (ψ), altitude (z), horizontal acceleration ($\dot{v}_x$), vertical acceleration ($\dot{v}_z$), horizontal speed (v_x), vertical speed (v_z), east and north ground speed (gs_{East}, gs_{North}), time of day (t), previous action (a_{t-1}) , and N_{wpt} future ownship relative waypoint positions ($\bar{x}_{\text{wpt}} - x$, $\bar{y}_{\text{wpt}} - y$). N_{wpt} is a hyperparameter that specifies how many future route segments to consider. The state space h for the i intruder aircraft available to the ownship at time t is defined as

$$h_t(i) = (\psi_{\text{rel}}^{(i)}, z_{\text{rel}}^{(i)}, \dot{v}_x^{(i)}, \dot{v}_z^{(i)}, v_x^{(i)}, v_z^{(i)}, gs_{\text{East}}^{(i)}, gs_{\text{North}}^{(i)}, \phi^{(i)}, d_o^{(i)}, a_{t-1}^{(i)}, x_{\text{wpt}}^{(i)}(j) - x, y_{\text{wpt}}^{(i)}(j) - y) \; \forall\, j \in [1, N_{\text{wpt}}].$$

The intruder state space contains information on the intruder aircraft similar to the ownship state with the intruder's acceleration, velocities, and previous action. It also includes ownship relative values of relative heading ($\psi_{rel}^{(i)}$), relative altitude ($z_{\text{rel}}^{(i)}$), relative bearing ($\phi^{(i)}$), the straightline distance between ownship and intruder ($d_o^{(i)}$), and the relative waypoint positions. With the state space specified, the components of the attention network can be defined as

$$\text{score}(s_t, \bar{h}_t) = s^\top W_1 \bar{h}_t \tag{4}$$

$$\eta_{s_t, \bar{h}_t} = \frac{\exp(\text{score}(s, \bar{h}_t))}{\sum_{j=1}^{n} \exp(\text{score}(s_t, \bar{h}_t^j))} \tag{5}$$

$$c_s = \sum_{i=1}^{n} \eta_{s, \bar{h}_t} \bar{h}_t^i \tag{6}$$

$$k_{s_t} = f(c_{s_t}) = \tanh(W_2 c_{s_t}), \tag{7}$$

where Luong's multiplicative style (Luong, Pham, and Manning 2015) is used as the score calculation. $\eta_{s_t, \bar{h}_t}$ is the attention weights of the ownship with respect to all of the other intruder aircraft, c_s is the context vector that represents the weighted contribution of the surrounding air traffic, and k_{s_t} is the attention vector that represents the abstract understanding of the surrounding air traffic. W_1 and W_2 are both learnable weight matrices determined through the neural network training. We then concatenate k_{s_t} with s_t to obtain the fixed length vector that can be passed through standard feed-forward layers of the actor and critic networks. Given that the attention network is operating as a state preprocessor, it can be used in both the actor and critic networks for SACD, or as part of a shared-layer network architecture.

Action Space

Actions for the agent reflect speed and altitude maneuvers, with a decision step of 4 seconds. The decision step is treated as a hyperparameter that can be modified based on the application. The action space is defined as

$$a_t = \{\dot{v}_{x_-}, \dot{v}_{x_0}, \dot{v}_{x_+}, v_{z_-}, v_{z_0}, v_{z_+}\}.$$

The available actions are decrease speed ($\dot{v}_{x_-}$), maintain current speed ($\dot{v}_{x_0}$), increase speed ($\dot{v}_{x_+}$), descend (v_{z_-}), maintain altitude (v_{z_0}), and climb (v_{z_+}). Given that there are multiple vertically stacked air corridors, climb and descend actions are halted when reaching a new lane. The magnitude of the speed and altitude changes are dependent on the performance envelope for a given aircraft type, provided by the OpenAP aircraft performance model (Sun, Hoekstra, and Ellerbroek 2020). Selected actions that result in speeds or altitudes outside of the performance envelope have no effect.

Parameter	Value
d_x^{NMAC}	500 feet
d_z^{NMAC}	100 feet
v_x : range	[5, 65] knots
z : range	[400, 1600] feet
N_{wpt}	5
d_o^{MAX}	3280 feet
γ	0.99
Reward coefficient χ	0.1
Reward coefficient δ	0.0001
Reward coefficient ϵ	0.001
Reward coefficient λ	0.01
Reward coefficient Ω	0.001
Replay memory capacity	8000000
Learning rate	5e-5
Batch size	512

Table 1: Finalized use-case hyperparameters.

Reward Function

In the context of separation assurance, the primary objective is to maintain a safe distance from intruder aircraft; however, operational suitability objectives (e.g., minimize maneuvers) are also important for real-world deployment. We achieve this objectives through defining the reward function as

$$R(s_t, h_t, a_t) = R(s_t, h_t) + R(a_t) - \Omega, \tag{8}$$

where $R(s_t, h_t)$ and $R(a_t)$ are defined as

$$R(s_t, h_t) = \begin{cases} \text{-1}, & \text{if } d_o^c < d_x^{\text{NMAC}} \\ & \text{and } z_{rel} < d_z^{\text{NMAC}} \\ -\chi + \delta \cdot d_o^c, & \text{if } d^{\text{NMAC}} \leq d_o^c < d^{\text{MAX}} \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

$$R(a_t) = \begin{cases} 0, & \text{if } a_t \in [\dot{v}_{x_0}, v_{z_0}] \\ -\epsilon, & \text{if } a_t \in [\dot{v}_{x_-}, \dot{v}_{x_+}] \\ -\lambda, & \text{if } a_t \in [v_{z_-}, v_{z_+}] \end{cases}. \tag{10}$$

In $R(s_t, h_t)$, d_o^c is the distance from the ownship to the closest intruder aircraft and d^{MAX} is the maximum distance to consider the closest intruder aircraft in the reward function. The hyperparameters χ and δ are small, positive constants to penalize aircraft as they approach the separation threshold, d^{NMAC}. In $R(a_t)$, ϵ and λ represent penalties for advisories that require a deviation from the aircraft's current speed or altitude, respectively, to encourage the agent to minimize maneuvering actions. Finally, in $R(s_t, h_t, a_t)$, the hyperparameter Ω represents a small, positive constant that is applied at every step in scenario. This parameter discourages aircraft from hovering, since slower aircraft will incur the Ω penalty for an extended time. Table 1 displays the finalized use-case hyperparameters. Each parameter was individually tuned by using a manual coordinate descent search.

Distributed Asynchronous Training

D2MAV-A introduced a distributed synchronous training procedure where parallel actors collect state-transition experience from the environment for a centralized learner to train on. Given that D2MAV-A is an on-policy RL algorithm based on Proximal Policy Optimization (Schulman et al. 2017), a synchronization step is required to ensure that each actor is collecting experience from the most up-to-date policy. SACD, an off-policy RL algorithm, does not require each actor to be collecting experience with the latest policy. Therefore, we can then decouple the algorithm training from the algorithm execution in the environment. In this way, the algorithm training can be performed asynchronously from the distributed actors to achieve high training throughput. Figure 1 illustrates the asynchronous centralized learning, decentralized execution scheme. We adopt the same network architecture for SACD as in (Christodoulou 2019) with the addition of the attention network for state preprocessing. The network layers consisted of 256 nodes for both the actor and critic networks. For target entropy annealing, we use the parameter values introduced in (Xu et al. 2021), with the exception of the standard deviation threshold, which we set to 0.07. Experiments were performed on the Lincoln Laboratory Supercomputer, consisting of 16 Intel Xeon Gold 6248 2.5 Ghz compute nodes with two NVIDIA Tesla V100 graphics processing units (GPUs) per compute node (Reuther et al. 2018).

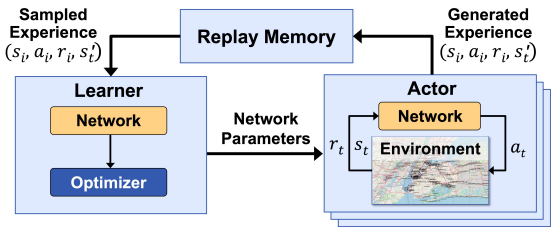

Figure 1: Distributed asynchronous training architecture.

Use-Case: Urban Air Corridors

Near-term AAM operations are expected to leverage existing visual flight rule (VFR) route networks (e.g., helicopter routes) as they will largely represent AAM corridors at lower air traffic densities (Federal Aviation Administration 2023). As such, we developed an environment based on the VFR route network and 29 vertiport locations for New York City as presented in (Alvarez et al. 2021) with the addition of vertically stacked lanes at 400, 700, 1000, 1300, and 1600 feet. The altitude lanes were selected to be representative of AAM flights; above the small aerial vehicle traffic expected below 400 ft (National Aeronautics and Space Administration 2020). The following subsections discuss the scenarios designed for this use-case, baselines, and experiment setup.

Scenario Design

A total of 20 days of representative AAM traffic is generated by UAMToolkit (Alvarez et al. 2021) with varying fleet sizes to provide a diverse set of operational densities. The AAM traffic demand is based on a displacement of 5% of the New York City taxi cab market, subject to the number of available AAM aircraft. The scenario generation takes into account

imperfect aircraft altitude by adding noise in the form of a uniform distribution ranging from -100 to 100 feet to offset the selected initial altitude. The fleet size determines how many aircraft are available to operate simultaneously; however, simultaneous operations will be limited based on the availability of vertiport parking spots. In this study, each vertiport was assumed to have four parking spots which results in a max of approximately 200 simultaneous operations. If the fleet size exceeds 200 aircraft, overall network utilization increases due to lower idle times by aircraft repositioning for passenger pickup. For this use-case, two aircraft performance models were chosen to simulate flights: (1) Eurocopter EC-135 and (2) surrogate AAM vehicle based on publicly available AAM aircraft specifications.

Baselines

Two baselines were used to benchmark the proposed SACD-A algorithm: (1) an unequipped agent that does not implement separation commands and (2) the D2MAV-A algorithm with speed and vertical lane change commands, an extension of the original speed-only D2MAV-A implementation.

Experiment Setup

Using the AAM-Gym testbed (Brittain et al. 2022) with 40 parallel workers for policy-rollout, each algorithm was trained for 10,000 iterations where one iteration is 64 environment steps (4 simulation seconds). Following training, evaluation was performed on 100 randomly sampled 3-hour windows of AAM operations for a given day. A sensitivity analysis was performed to understand how robust the algorithms are to state observation noise and fleet size by simulating various fleet sizes with both automatic dependent surveillance-broadcast (ADS-B) noise and perfect surveillance. ADS-B noise is applied as a Gaussian distribution over the latitude, longitude, and altitude, with a standard deviation 0.0001 for latitude and longitude, and 100 feet for altitude. To test the robustness of the algorithms under various levels of uncertainty, we performed a sweep over two stressing parameters: (1) probability of communication and (2) probability of policy equipage with a fleet size of 100 aircraft. Probability of communication refers to the ability of a given aircraft to receive intruder state information. This condition is applied at aircraft initialization such that an aircraft without communication does not observe the intruder aircraft for the entire flight. Probability of policy equipage represents the likelihood that an aircraft is equipped with a separation assurance logic. Policy equipage is determined at aircraft initialization and aircraft not equipped with the logic follow their original flight altitude and speeds. In situations containing unequipped aircraft, the algorithms must learn and adapt to the non-cooperative aircraft.

Results

We adopt the use of the risk ratio as the primary metric for evaluating the safety of the algorithms. Risk ratio is a commonly used metric for aircraft collision avoidance systems (Alvarez et al. 2019) and is defined as

$$\text{risk ratio} = \frac{P(\text{NMAC})_{\text{logic}}}{P(\text{NMAC})_{\text{no logic}}}. \tag{11}$$

The risk ratio provides a measure of how much safety improvement can be achieved using an algorithm compared to the case when no logic is present (unequipped aircraft). Values close to zero represent that the algorithms resolved all NMAC events, whereas values greater than one indicate the algorithms results in more NMACs than the unequipped aircraft. Risk ratio equal to one indicates no safety improvement with the algorithms.

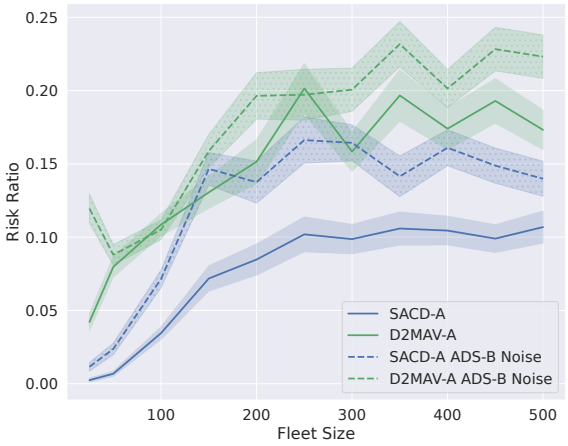

Figure 2: Risk ratio versus fleet size (shaded regions represent standard deviation).

Figure 2 shows the algorithm risk ratio for various fleet sizes and surveillance sources. Both algorithms were able to reduce airspace risk over the unequipped agent given risk ratio values less than one. For all fleet sizes SACD-A outperforms D2MAV-A, achieving a steady-state risk ratio at fleet size = 250. Interestingly, SACD-A with ADS-B noise outperforms D2MAV-A with perfect surveillance, demonstrating the robustness of SACD-A.

Figure 3 shows the algorithm risk ratio for various communication probabilities. As expected, risk ratio increases as the probability of communication decreases since fewer agents are able to receive intruder state information. However, it is seen that SACD-A achieves a much smaller risk ratio over D2MAV-A, with the difference becoming more significant as the probability of communication decreases.

The impact of the probability of equipage is shown in Figure 4. At $P(\text{equipped}) \leq 0.75$, D2MAV-A outperforms SACD-A, indicating that D2MAV-A may perform better at adapting to the behavior of unequipped agents. However, when the majority of aircraft are logic-equipped ($P(\text{equipped}) > 0.75$) SACD-A significantly outperforms D2MAV-A, resulting in a risk ratio of 0.05 when at least 95% of aircraft are equipped with the SACD-A policy.

Apart from safety, it is also important to consider the operational suitability of the learned policy (e.g., reduce maneuvers). Table 2 shows the probability distribution of air-

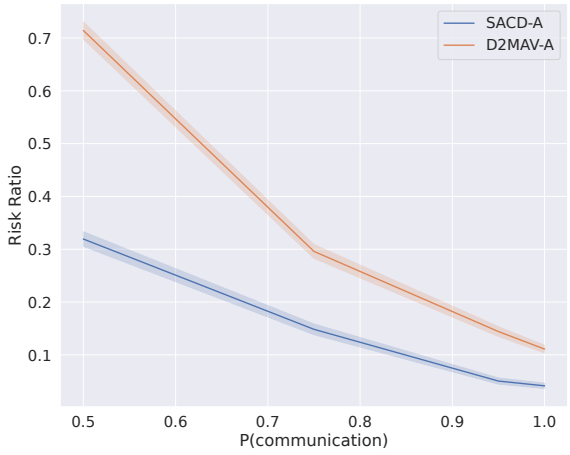

Figure 3: Risk ratio versus probability of communication (shaded regions represent standard deviation).

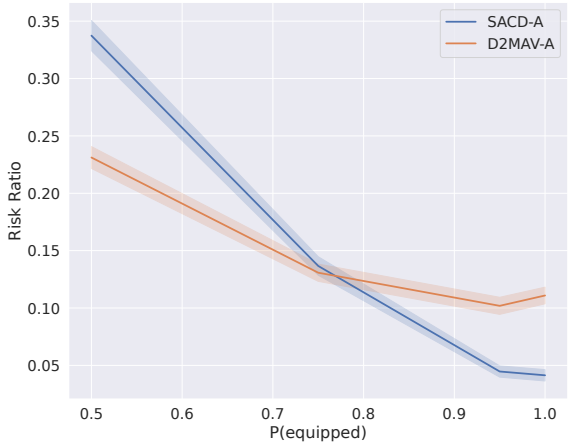

Figure 4: Risk ratio versus probability of equipped (shaded regions represent standard deviation).

craft maneuvers for the SACD-A algorithm from the evaluation scenarios. It is seen that the majority of actions selected by the agent are non-maneuvering actions including 'maintain speed' ($\dot{v}_{x_0}$) and 'maintain altitude' (v_{z_0}), successfully achieving our objective of minimizing maneuvering to only when it is necessary. The impact of the step penalty is reflected in the percentage of 'speed up' ($\dot{v}_{x_+}$) actions, given the agent is encouraged to navigate the route as quickly as possible. Due to a slight skew towards lower altitudes in the generated scenarios, the 'climb' action is preferred since it allows for higher separation through climbing.

The algorithm sample efficiency is compared over the 10,000 training iterations. SACD-A trained on 2.36 billion transitions, while D2MAV-A trained on 256 million transitions, an almost 10x increase. This shows that the distributed asynchronous training architecture for SACD-A provides increased training throughput compared to D2MAV-A.

a	$\dot{v}_{x_{(-)}}$	$\dot{v}_{x_{(0)}}$	$\dot{v}_{x_{(+)}}$	$v_{z_{(-)}}$	$v_{z_{(0)}}$	$v_{z_{(+)}}$
$P(a)$	0.005	0.422	0.05	0.028	0.488	0.007

Table 2: SACD-A action distribution throughout evaluation.

Discussion

This work provides a promising solution to increase the safety and efficiency of air transportation. When considering a safety-critical application such as separation assurance, additional steps would need to be performed to transition this approach to real-world deployment. The simulation environment would first need to introduce additional real-world models, such as latency, weather, and live air traffic using data from sources like the FAA System-Wide Information Management system and the MIT Lincoln Laboratory Corridor Integrated Weather System. Then, studies will need to be conducted to evaluate the interoperability with existing systems such as the next generation collision avoidance system (ACAS X). Deployment of the proposed system can leverage the approval process set by systems such as ACAS X, which underwent multi-organization validation through RTCA, EUROCONTROL, and the FAA. We believe a similar approach could be adapted to support the proposed system where an RTCA committee is formed and an iterative development process is conducted with the simulation environment parameters, evaluation data, and algorithm performance agreed upon by the committee. An in-depth understanding of the learning-based system's behavior will require a broad set of validation and verification techniques such as adaptive stress testing to identify failures, Monte Carlo large scale evaluations, real world testing, and formal methods. Near-term deployment could be achieved through human supervisory control by the algorithm instead acting as a recommender system to provide air traffic control a recommended maneuver to resolve a potential conflict or improve efficiency. Further simulation studies and human-in-the-loop experiments need to be conducted to understand the impact of real-world implementation.

Conclusions

A decentralized reinforcement learning framework is introduced to safely and efficiently separate aircraft in high density AAM corridors through speed and vertical maneuvers. A rigorous set of numerical experiments demonstrate the effectiveness of the SACD-A framework over existing approaches by introducing sources of uncertainty and high traffic density scenarios. The operational suitability of the proposed framework is shown through maximizing non-maneuvering actions, so that actions are selected only if necessary to resolve a conflict or increase efficiency. Looking forward, we to plan apply the SACD-A framework to more complex scenarios as well as to explore the framework interoperability with existing airspace deconfliction systems.

References

Alvarez, L. E.; Jessen, I.; Owen, M. P.; Silbermann, J.; and Wood, P. 2019. ACAS sXu: Robust Decentralized Detect and Avoid for

Small Unmanned Aircraft Systems. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 1–9.

Alvarez, L. E.; Jones, J. C.; Bryan, A.; and Weinert, A. J. 2021. Demand and Capacity Modeling for Advanced Air Mobility. In *AIAA AVIATION 2021 FORUM*.

Amato, C.; and Shani, G. 2010. High-Level Reinforcement Learning in Strategy Games. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, 7582. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9780982657119.

Badea, C.; Groot, D.; Veytia, A. M.; Ribeiro, M.; Ellerbroek, J.; Hoekstra, J.; and Dalmau, R. 2022. Lateral and Vertical Air Traffic Control Under Uncertainty Using Reinforcement Learning. In *12th SESAR Innovation Days*.

Brittain, M.; Alvarez, L. E.; Breeden, K.; and Jessen, I. 2022. AAM-Gym: Artificial Intelligence Testbed for Advanced Air Mobility. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, 1–10.

Brittain, M.; and Wei, P. 2018. Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning. In *2018 International Conference on Research in Air Transportation (ICRAT)*.

Brittain, M.; and Wei, P. 2019. Autonomous Separation Assurance in An High-Density En Route Sector: A Deep Multi-Agent Reinforcement Learning Approach. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 3256–3262.

Brittain, M.; and Wei, P. 2022. Scalable Autonomous Separation Assurance With Heterogeneous Multi-Agent Reinforcement Learning. *IEEE Transactions on Automation Science and Engineering*, 1–12.

Brittain, M. W.; and Wei, P. 2021. One to Any: Distributed Conflict Resolution with Deep Multi-Agent Reinforcement Learning and Long Short-Term Memory. In *AIAA Scitech 2021 Forum*.

Brittain, M. W.; Yang, X.; and Wei, P. 2021. Autonomous Separation Assurance with Deep Multi-Agent Reinforcement Learning. *Journal of Aerospace Information Systems*, 18(12): 890–905.

Christodoulou, P. 2019. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*.

Federal Aviation Administration. 2023. Urban Air Mobility Concept of Operations version 2.0. https://www.faa.gov/sites/faa.gov/files/Urban%20Air%20Mobility%20%28UAM%29%20Concept%20of%20Operations%202.0_0.pdf. Accessed on 2023-10-04.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018a. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1861–1870. PMLR.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018b. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hoekstra, J.; and Ellerbroek, J. 2016. BlueSky ATC Simulator Project: an Open Data and Open Source Approach. In *Proceedings of the seventh International Conference for Research on Air Transport (ICRAT)*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

National Academies of Sciences, Engineering, and Medicine. 2020. *Advancing Aerial Mobility: A National Blueprint*. Washington, DC: The National Academies Press. ISBN 978-0-309-67026-5.

National Aeronautics and Space Administration. 2020. National Aeronautics and Space Administration (NASA) UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML)4. https://ntrs.nasa.gov/api/citations/20205011091/downloads/UAM%20Vision%20Concept%20of%20Operations%20UML-4%20v1.0.pdf. Accessed on 2023-10-04.

Pham, D.-T.; Tran, N. P.; Alam, S.; Duong, V.; and Delahaye, D. 2019. A Machine Learning Approach for Conflict Resolution in Dense Traffic Scenarios with Uncertainties. In *13th USA/Europe ATM R&D Seminar*.

Reuther, A.; Kepner, J.; Byun, C.; Samsi, S.; Arcand, W.; Bestor, D.; Bergeron, B.; Gadepally, V.; Houle, M.; Hubbell, M.; Jones, M.; Klein, A.; Milechin, L.; Mullen, J.; Prout, A.; Rosa, A.; Yee, C.; and Michaleas, P. 2018. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 1–6. IEEE.

Ribeiro, M.; Ellerbroek, J.; and Hoekstra, J. 2020. Improvement of Conflict Detection and Resolution at High Densities Through Reinforcement Learning. In *Proceedings of the International Conference for Research in Air Transportation*.

Ribeiro, M.; Ellerbroek, J.; and Hoekstra, J. 2022. Improving Algorithm Conflict Resolution Manoeuvres with Reinforcement Learning. *Aerospace*, 9(12): 847.

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

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Sun, J.; Hoekstra, J. M.; and Ellerbroek, J. 2020. OpenAP: An open-source aircraft performance model for air transportation studies and simulations. *Aerospace*, 7(8): 104.

Tran, P. N.; Pham, D.-T.; Goh, S. K.; Alam, S.; and Duong, V. 2020. An Interactive Conflict Solver for Learning Air Traffic Conflict Resolutions. *Journal of Aerospace Information Systems*, 17(6): 271–277.

Vinyals, O.; Ewalds, T.; Bartunov, S.; Georgiev, P.; Vezhnevets, A. S.; Yeo, M.; Makhzani, A.; Küttler, H.; Agapiou, J.; Schrittwieser, J.; Quan, J.; Gaffney, S.; Petersen, S.; Simonyan, K.; Schaul, T.; van Hasselt, H.; Silver, D.; Lillicrap, T.; Calderone, K.; Keet, P.; Brunasso, A.; Lawrence, D.; Ekermo, A.; Repp, J.; and Tsing, R. 2017. Starcraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8: 279–292.

Xu, Y.; Hu, D.; Liang, L.; McAleer, S.; Abbeel, P.; and Fox, R. 2021. Target entropy annealing for discrete soft actor-critic. *arXiv preprint arXiv:2112.02852*.