Труды МАИ. 2025. № 144
Trudy MAI. 2025. No. 144. (In Russ.)

ИНФОРМАТИКА, МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ

Научная статья
УДК 517.977.5
URL: https://trudymai.ru/published.php?ID=186320
EDN: https://www.elibrary.ru/SRPYLU

УЧЁТ РЕАКЦИИ ПИЛОТА В АЛГОРИТМЕ ПРЕДОТВРАЩЕНИЯ ВОЗДУШНЫХ СТОЛКНОВЕНИЙ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Евгений Сергеевич Неретин[1]✉, Лю Цзочэн[2]

[1]филиал ПАО "Яковлев" - Центр комплексирования

Москва, Россия

[2]Северо-Западный политехнический университет

Сиань, Китай

[1]✉evgeny.neretin@ic.yakovlev.ru

Аннотация. Текущие системы предотвращения воздушных столкновений используют псевдокод или числовые таблицы для представления оптимальных стратегий, обеспечивая высокий уровень безопасности полетов. Однако ряд проблем в процессе разработки этих систем ограничил интеграцию систем предотвращения столкновений с авионикой и дальнейшее развитие в будущем. Эти проблемы включают неточности, вызванные интерполяционными методами, игнорирование изменчивости реакции пилота и создание слишком больших числовых таблиц для

хранения оптимальных стратегий. В ответ на эти вызовы мы используем методы глубокого обучения с подкреплением (Deep Reinforcement Learning, DRL) для решения проблемы предотвращения столкновений и предлагаем более принципиальный подход, использующий вероятностную модель для учета изменчивости реакции пилота вблизи. В данной работе мы сначала представили текущий статус исследований систем предотвращения воздушных столкновений и соответствующие теории глубокого обучения с подкреплением. Затем мы создали симуляционную среду, адаптированную к проблемам предотвращения столкновений самолетов, разработали систему вознаграждений, изменяющуюся в зависимости от времени до конфликта (Time to Conflict, TTC), и создали вероятностную модель для учета изменчивости реакции пилота. Мы применили алгоритм DQN (Deep Q-Network) для обучения агента, способного решать проблемы предотвращения столкновений самолетов, а также учитывать координацию высоты двух самолетов. Наконец, мы протестировали эффективность и надежность алгоритма с помощью симуляционных экспериментов в сценариях предотвращения столкновений различной сложности.

Ключевые слова: реакция пилота, глубокое обучение с подкреплением, воздушное столкновение, Марковский процесс принятия решений, Динамическое программирование

***Для цитирования*:** Неретин Е.С., Лю Ц. Учёт реакции пилота в алгоритме предотвращения воздушных столкновений на основе глубокого обучения с подкреплением // Труды МАИ. 2025. № 144. URL: https://trudymai.ru/published.php?ID=186320

Original article

ADDRESSING PILOT RESPOND IN AIRBORNE COLLISION AVOIDANCE ALGORITHM BASED ON DEEP REINFORCEMENT LEARNING

Neretin E.S., Zuocheng L.

[1]Branch of PJSC Yakovlev – integration center

Moscow, Russia

[2]Northwestern Polytechnical University

Xi'An, P. R. China

[1]✉evgeny.neretin@ic.yakovlev.ru

Abstract. Current airborne collision avoidance systems use pseudocode or numerical tables to represent optimal strategies, achieving a high level of flight safety. However, a series of issues during the development of these systems have limited the integration of collision avoidance systems with avionics and further development in the future. These issues include inaccuracies caused by interpolation techniques, neglect of pilot response variability, and the generation of too large numerical tables used to store optimal strategies. In response to these challenges, we employ Deep Reinforcement Learning (DRL) methods to solve the collision avoidance problem and pursue a more principled approach that uses a probabilistic model to account for pilot response variability in close proximity. In this paper, we first introduced the current research status of airborne collision avoidance systems and the relevant theories of deep reinforcement learning. We then constructed a simulation

environment tailored to aircraft collision avoidance problems, developed a reward system that varies with the Time to Conflict (TTC), and established a probabilistic model to handle pilot response variability. We applied the DQN (Deep Q-Network) algorithm to train an agent capable of addressing aircraft collision avoidance problems while also considering altitude coordination of the two aircrafts. Finally, we tested the algorithm's effectiveness and robustness through simulation experiments in collision avoidance scenarios of varying difficulty.

Keywords: Pilot respond, Deep reinforcement learning, airborne collision avoidance, Markov decision-making process, Dynamic Programming

***For citation*:** Neretin E.S., Zuocheng L. Addressing pilot respond in airborne collision avoidance algorithm based on deep reinforcement learning // Trudy MAI. 2025. No. 144. (In Russ.) URL: https://trudymai.ru/published.php?ID=186320

Введение

Система предупреждения столкновений и предотвращения столкновений в воздухе (TCAS) использует бортовой радиолокационный маяк для мониторинга местного воздушного движения с целью снижения риска столкновений в воздухе. Итеративный процесс разработки TCAS включает использование псевдокода для задания логики, который содержит множество эвристических правил и настроек параметров, и все более неспособен справляться с будущими сложными воздушными пространствами [1]. Некоторые недавние исследования сосредоточены на разработке систем предотвращения столкновений для преодоления ограничений существующего

оборудования и программного обеспечения [2]. Например, предлагается система предотвращения столкновений самолетов X (ACAS X) на основе марковского процесса принятия решений (MDP) как метод для предотвращения столкновений в будущем [3]. ACAS X использует итерацию значений для решения оптимальной стратегии в процессе разработки и использует числовую таблицу поиска для её представления, что обеспечивает высокий уровень безопасности [4] [5].

Однако в процессе разработки таблицы поиска ACAS X часто требуют сотен гигабайт хранения с плавающей запятой, что накладывает огромные ограничения на использование ACAS X в авионике [4, 6]. Простая техника для уменьшения размера таблицы оценок заключается в уменьшении масштаба таблицы после динамического программирования (DP). Чтобы минимизировать снижение качества решений, состояния в таблице удаляются из областей, где изменения между значениями были плавными. Другие методы включают использование глубоких нейронных сетей для аппроксимации сжатия таблицы [6] и представление дискретных числовых таблиц в виде гауссовских процессов [7] или kd-деревьев [8] для устранения избыточности в таблице. После получения числовой таблицы система ACAS должна быть проверена в замкнутой модели, включая динамику системы, чтобы обеспечить безопасность разработанной системы [9, 10, 11, 12]. Недостатком такого процесса разработки является то, что процессы разработки и оценки разделены, и проблемы в модели необходимо вручную исправлять после оценки, что не способствует последующему обслуживанию.

В процессе разработки ACAS X проблема предупреждения столкновений

рассматривалась как модельная проблема, где награды и матрицы перехода состояний были определены в рамках динамической модели. Однако в реальных сценариях столкновений переходы состояний и награды неизвестны. Использование интерполяционных техник для дискретизации непрерывного пространства может привести к неточностям в модели. Поэтому можно рассматривать проблему предупреждения столкновений как проблему без модели. В этом подходе агент взаимодействует непосредственно с окружающей средой, а среда отвечает на действия агента, предоставляя следующее действие и соответствующую награду. Оценка стратегии становится более разумной за счет выборки траекторий взаимодействия агент-среда.

Еще одной проблемой является реакция пилота и надежность системы. Как и TCAS, ACAS X не управляет самолетом напрямую; он может только выдавать рекомендации пилотам о том, как маневрировать по вертикали, чтобы предотвратить столкновение. Как показали данные радара, около 20% рекомендаций систем предотвращения столкновений не принимаются в Европе, и когда пилоты следуют рекомендациям, их действия часто не так сильны, как рекомендовано [2, 13]. Для обеспечения безопасности логика предотвращения столкновений должна учитывать изменчивость реакции. Текущая логика TCAS явно не моделирует изменчивость реакции пилота. Вместо этого используется детерминированная модель для прогнозирования будущих траекторий самолета и применяется сложный набор эвристик для обеспечения устойчивости к непредвиденному поведению. В процессе разработки будущих версий ACAS X уже начаты усилия по учету воздействия

реакции пилота [14].

Таким образом, в этой статье мы используем методы глубокого обучения с подкреплением для решения проблемы предотвращения столкновений самолетов и предлагаем более принципиальный подход, использующий вероятностную модель для учета изменчивости реакции пилота вблизи. Модели сети, полученные с помощью глубокого обучения с подкреплением, могут значительно сократить объем памяти и позволить системе обновляться самостоятельно в процессе взаимодействия с окружающей средой, что сокращает процесс разработки. Система предотвращения столкновений, использующая вероятностную модель реакции, может лучше отражать реальные условия и является более надежной по сравнению с системой, предполагающей детерминированную реакцию.

1. Глубокое обучение с подкреплением

A. Марковский процесс принятия решений (MDP)

MDP – это модельная структура для принятия решений, которая в настоящее время широко используется в исследованиях в области финансов, проектирования систем автоматизации роботов и других областях. MDP формулируется как кортеж (S, A, R, T), где $s_t \in S$ – это состояние в данный момент времени t, $a_t \in A$ – действие, предпринимаемое агентом в момент времени t в результате процесса принятия решения, $r_t = R(s_t, a_t, s_{t+1})$ – награда, полученная агентом в результате выполнения действия a_t из состояния s_t и перехода в состояние s_{t+1}, и $T(s_t , a, s_t+1)$ – функция перехода, описывающая динамику среды и отображающая вероятность $p(s_{t+1} \mid s_t, a_t)$

перехода в состояние s_{t+1} при выполнении действия a_t из состояния s_t [15].

MDP требует, чтобы проблема моделирования была марковской, то есть переход состояния в состояние s_{t+1} должен зависеть от текущего состояния s_t и выполненного действия a. Марковское предположение обычно справедливо, если в состоянии включено достаточно информации о проблеме [4].

Решение MDP называется оптимальной политикой π^*, которая определяет оптимальное действие $a^* \in A$, которое можно предпринять из каждого состояния $s \in S$ для максимизации ожидаемой будущей награды. Из этой оптимальной политики π^* можно вычислить оптимальную функцию ценности $V^*(s)$, которая описывает максимальное ожидаемое значение, которое можно получить из каждого состояния $s \in S$.

Решение MDP означает нахождение оптимальной политики, которая максимизирует ожидаемое значение оптимальной функции ценности. Традиционный метод решения MDP заключается в использовании метода итерации значений для решения уравнения Беллмана (1) во всем пространстве состояний.

$$V_k(s_t) = \max_a \left[r(s_t, a_t) + \gamma \sum_{s_{t+1}} T(s_{t+1} | s_t, a_t) V_{k-1}(s_{t+1}) \right] \quad (1)$$

Дисконтирующий фактор γ служит для балансирования между немедленной наградой и будущей наградой. Малые значения γ, близкие к нулю, отдают предпочтение немедленным наградам, тогда как большие значения γ, близкие к единице, отдают предпочтение долгосрочным наградам.

B. Динамическое программирование, Q-learning и глубокое Q-learning

После моделирования задачи как MDP следующим шагом является решение MDP. Динамическое программирование — это метод на основе модели, который итеративно обновляет функцию ценности состояния или функцию ценности состояния-действия MDP, чтобы в конечном итоге получить оптимальную стратегию. Итерация ценности приближается к оптимальной функции ценности, итеративно обновляя функцию ценности состояния для получения оптимальной стратегии; в то время как итерация стратегии приближается к оптимальной стратегии, итеративно обновляя саму стратегию. Другой метод заключается в том, чтобы изучить оптимальную стратегию, постоянно пробуя различные действия в MDP и корректируя функцию ценности состояния-действия (то есть функцию Q-ценности) в зависимости от полученной награды. Q-learning — это метод обучения с подкреплением, не основанный на модели, а на опыте, который не требует моделирования окружающей среды, а учится напрямую через взаимодействие с ней.

Мы определяем дискретные временные шаги следующим образом: t=1, 2, 3 и т.д., где на каждом временном шаге t агент взаимодействует с окружающей средой, чтобы получить состояние окружающей среды s_t. Затем агент выбирает действие a_t на основе текущего состояния s_t, где $a_t \in a(s_t)$, а $a(s_t)$ — это набор действий, которые можно выбрать в состоянии s_t. Далее агент выполняет действие a_t, и состояние окружающей среды переходит в s_{t+1}. В то же время агент получает награду r_t. Этот процесс повторяется непрерывно, пока агент не достигнет конечного состояния. Предположим, что продолжительность одного эпизода составляет от времени t до T; следовательно,

накопленная награда, полученная агентом за это время, будет следующей:

$$R_t = \sum_{t'-t}^{T} \gamma^{t'-t} r_{t'} \tag{2}$$

Где γ является коэффициентом дисконтирования, и его диапазон значений составляет [0, 1]. Роль коэффициента дисконтирования заключается в том, чтобы взвесить влияние будущих наград на совокупную награду. Функция ценности действия определяется следующим образом:

$$Q(s,a) = E\left[s_t = s, a_t = a, \pi\right] \tag{3}$$

Уравнение (3) представляет собой накопленную награду, которую агент может получить, если он выполняет действие *a* в состоянии *s* и всегда следует политике π до конца эпизода. Формула обновления функции ценности действия определяется следующим образом:

$$Q(s,a) \leftarrow Q^{\pi}\left(s,a\right) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{4}$$

Используя уравнение (4) для итераций, в конечном итоге получается оптимальная функция ценности действия Q-learning.

Чтобы расширить область применения Q-learning, Мних и др. предложили модель DQN, объединяющую сверточную нейронную сеть с Q-learning [16, 17]. DQN имеет две сети с одинаковыми гиперпараметрами. Оценочная сеть использует $Q(s, a; \theta)$ в качестве Q-функции для аппроксимации функции ценности действия, в то время как целевая сеть использует $Q(s, a; \theta^{-})$. Параметры оценочной сети обновляются в режиме реального времени, и их копии передаются в целевую сеть каждые *N* итераций. Когда количество итераций равно *i*, функция потерь выглядит следующим образом:

$$L_i(\theta_i) = E_{(s,a,r,s')}\left[(y_i^{DQN} - Q(s,a;\theta))^2\right]$$
$$y_i^{DQN} = r + \gamma \max_{a'} Q(s',a';\theta^-) \quad (5)$$

где θ_i представляет собой параметры сети в оценочной сети, а θ_i^- представляет собой параметры в целевой сети. DQN и сеть изображены на Рисунке 1.

Основные особенности DQN следующие: 1) DQN является методом обучения end-to-end, который использует глубокие нейронные сети для извлечения признаков и обучение с подкреплением для адаптации к новым средам [18]. 2) DQN применяет технологию воспроизведения опыта (experience replay) для повышения эффективности данных путем выборки исторических данных и уменьшения корреляции данных [19]. 3) Целевая и оценочная сети используют одни и те же гиперпараметры, демонстрируя универсальность алгоритма для различных задач.

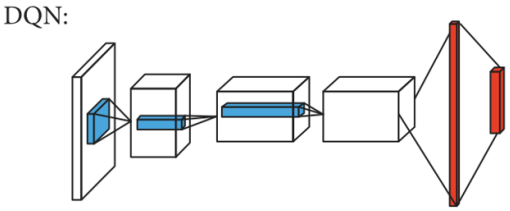

Рис. 1. Структуры сети DQN

2. Предотвращение столкновений с учетом модели реакции пилота с использованием DRL

A. Определение пространства состояний и действий

TCAS – это бортовая система, предоставляющая советы по трафику для предотвращения близких столкновений в воздухе (NMAC) с другим самолетом. NMAC определяются как разделение менее 100 футов по вертикали и 500 футов по

горизонтали. TCAS I предлагает консультации по трафику (TA) для предупреждения о близких самолетах, в то время как TCAS II, используемый большинством коммерческих самолетов, предоставляет как TA, так и консультации по разрешению конфликтов (RA). TA предупреждает о потенциальных конфликтах, в то время как RA предлагает пилотам вертикальные маневры (набор высоты или снижение) для их предотвращения, с возможной координацией между самолетами, оборудованными TCAS II [20].

Для поддержания согласованности с TCAS и ACAS X, в данной статье акцент делается на вертикальном избегании столкновений. Пространство состояний для задачи предотвращения столкновений состоит из пяти переменных. Таблица 1 описывает эти переменные и их диапазоны, а на Рисунке. 2 представлено их визуальное отображение. Первые три переменные описывают относительные положения и вертикальные скорости собственного и вторгающегося самолета. Четвертая переменная, τ, обобщает горизонтальную геометрию, указывая время до того, как горизонтальное расстояние между двумя самолетами станет менее 500 футов, и это также определяется временем до конфликта (Time to Conflict, TTC). Включение предыдущего совета в пространство состояний позволяет нам штрафовать за изменения или усиления советов, сохраняя марковское свойство.

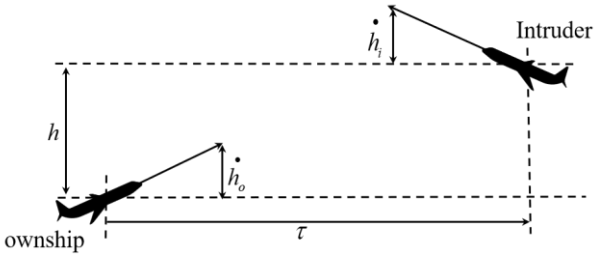

Рис. 2 Визуальное представление переменных состояния

Таблица 1.Переменные пространства состояний

Переменная	**Описание**	**Значения**	**Единицы измерения**
h	Относительная высота нарушителя	[-2500, 2500]	ft
$\dot{h}_o$	Вертикальная скорость собственного самолета	[-70, 70]	ft/s
$\dot{h}_i$	Вертикальная скорость нарушителя	[-50, 50]	ft/s
τ	Время до потери горизонтального разделения (TTC)	[0, 40]	s
a_{prev}	Предыдущий совет	См. Таблица 2	-

Таблица 2. Набор советов

Действие	**Описание**	**Ускорение**
COC	Нет конфликта	0
DES1500	Спуск<=-25ft/s	-g/3
CL1500	Взлет>=25ft/s	g/3
SDES1500	Спуск<=-25ft/s	-g/2.5
SCL1500	Взлет>=25ft/s	g/2.5
SDES2500	Спуск<=-42ft/s	-g/2.5
SCL2500	Взлет>=42ft/s	g/2.5

Таблица 3. Доступность советов

Действие	**Доступно от**
COC	В любое время
DES1500	COC
CL1500	COC
SDES1500	CL1500, SCL1500, SCL2500, SDE2500
SCL1500	DES1500, SDE1500, SDES2500, SCL1500
SDES2500	DES1500, SDES1500
SCL2500	CL1500, SCL1500

Пространство действий включает советы, которые система предотвращения столкновений может предоставить во время полета, всего 7 возможных советов, как показано в Таблице 2. Все советы, кроме COC, вызывают предупреждение и направляют самолет в определенный диапазон вертикальных скоростей с

соответствующим ускорением. Совет COC указывает на отсутствие немедленной угрозы столкновения с вторгающимся самолетом.

Таблица 3 описывает доступность каждого совета в зависимости от текущего отображаемого совета. Например, COC может быть выдан в любое время. Однако DES1500 и CL1500, будучи начальными советами, могут быть выданы только если в данный момент пилоту отображается COC. SDES1500 может быть выдан после CL1500, SCL1500 и SCL2500, выступая как разворот, или после SDES2500, выступая как ослабление. Важно отметить, что SDES1500 не может следовать за COC и также не может следовать за DES1500 из-за их сходства по своей природе. Таким образом, исходя из доступности каждого совета под текущим советом, агент на самом деле может выбирать только из трех действий в каждом состоянии.

B. Динамическая модель самолета

Динамическая модель может быть записана как уравнение (6). Мы предполагаем временной шаг в одну секунду, что означает обновление системы предотвращения столкновений с частотой 1 Гц. Для усложнения симуляционной среды мы ограничиваем диапазон ускорений вторгающегося самолета значениями от [$-a_{\text{int}}$, a_{int}], и предполагаем, что скорость вторгающегося самолета h_{int} изменяется в направлении состояния столкновения на каждом шаге.

$$\begin{bmatrix} h \\ \dot{h}_{own} \\ \dot{h}_{\text{int}} \\ \tau \\ a_{prev} \end{bmatrix} \leftarrow \begin{bmatrix} h + \dot{h}_{\text{int}} + 0.5\ddot{h}_{\text{int}} - \dot{h}_{own} - 0.5\ddot{h}_{own} \\ \dot{h}_{own} + \ddot{h}_{own} \\ \dot{h}_{\text{int}} + \ddot{h}_{\text{int}} \\ \tau - 1 \\ a'_{prev} \end{bmatrix} \tag{6}$$

C. Вероятностная модель реакции пилота

Когда выдается совет, на следующем временном шаге собственный самолет следует ускорениям, основанным на этом совете, с соответствующими вероятностями, показанными в Таблице 4. Эта модель реакции включает две особенности, которые повышают устойчивость динамической модели. Во-первых, она предполагает, что собственный самолет будет следовать ускорениям, связанным с его предыдущим советом, а не текущим, учитывая кратковременную задержку пилота в ответ на совет. Во-вторых, она учитывает возможные ошибки в реакции самолета, предполагая, что собственный самолет будет ускоряться в противоположном направлении от своего совета в 20% случаев. Этот сценарий особенно актуален, когда за выполнение маневра предотвращения столкновения отвечает человеческий пилот, так как он может не сразу реагировать на совет и потенциально может ускоряться против него.

Таблица 4. Рекомендации с использованием вероятностной модели ответа пилота

Действие	**Вероятности Ответа Пилота**	**Ускорение**
COC	[0.34, 0.33, 0.33]	[0, -g/3, g/3]
DES1500	[0.5, 0.3, 0.2]	[-g/3, -g/2, g/3]
CL1500	[0.5, 0.3, 0.2]	[g/3, g/2, -g/3]
SDES1500	[0.5, 0.3, 0.2]	[-g/2.5, -g/2, g/3]
SCL1500	[0.5, 0.3, 0.2]	[g/2.5, g/2, -g/3]
SDES2500	[0.5, 0.3, 0.2]	[-g/2.5, -g/2, g/3]
SCL2500	[0.5, 0.3, 0.2]	[g/2.5, g/2, -g/3]

D. Формирование награды

В процессе обучения с подкреплением (RL) дизайн функции награды имеет решающее значение, поскольку награда $r(s, a)$ служит единственным критерием для оценки качества выбранного действия a в текущем состоянии s. Плохо спроектированные функции награды могут привести к несходимости всего алгоритма.

В задачах предотвращения столкновений функция награды должна балансировать между безопасностью и эффективностью. Поэтому наша цель – предотвратить столкновения, минимизируя при этом выдачу отвлекающих предупреждений и неожиданных действий системы (таких как усиление рекомендаций, изменение рекомендаций или создание пересечения высот между самолетами). Кроме того, чтобы минимизировать влияние на другое воздушное пространство во время разрешения конфликта, мы поощряем весь процесс разрешения оставаться в определенном диапазоне высот. Таким образом, дизайн функции награды в первую очередь разделен на две части. Во-первых, в конце TTC нам необходимо, чтобы относительная высота самолетов поддерживалась на определенной высоте h_{rel}, и штрафы налагаются за другие состояния, такие как столкновение, чрезмерно высокая или низкая относительная высота. Первая часть функции награды может быть сформулирована следующим образом:

$$\begin{aligned} R_1 = &-\omega_{NMAC} 1\{|h_{rel}| \le h_{col}\} + \omega_{NMAC}\left(\frac{|h_{rel}| - h_{col}}{h_{rel\_s} - h_{col}}\right)\{h_{col} \le |h_{rel}| < h_{rel\_s}\} \\ &+ \max\left(-2\omega_{leav}\left(|h_{rel}| - h_{rel\_s}\right), -25\right)\{|h_{rel}| \ge h_{rel\_s}\} \end{aligned} \tag{7}$$

Где w_{NMAC} штрафует за столкновение между собственным самолетом и нарушителем, h_{rel} представляет относительную высоту между двумя самолетами, h_{col}

указывает высоту, на которой происходит конфликт между двумя самолетами, h_{rel_s} обозначает желаемую относительную высоту, которую мы хотим, чтобы два самолета достигли, а w_{leav} представляет штраф, накладываемый, когда относительная высота между двумя самолетами чрезмерно велика.

В течение периода TTC нам необходимо обращать внимание на высоту собственного самолета, изменения в статусе предупреждения и ситуацию пересечения высот между двумя самолетами. Функция награды может быть сформулирована следующим образом:

$$R_2 = -\omega_{leav} 1\left\{\left|h_{own}\right| > h_{upper}\right\} - \left(\omega_{alert} + \omega_{reversal} + \omega_{strength} + \omega_{crossing}\right)\exp\left(t - tau\right) + \omega_{COC} \quad (8)$$

w_{leav} представляет штраф, накладываемый, когда высота собственного самолета (h_{own}) превышает заданный верхний предел диапазона высот (h_{upper}). w_{Alert}, w_{stren}, $w_{reversal}$, w_{cros} штрафуют систему за выдачу предупреждений и ложных тревог. Среди них, τ представляет Время до Столкновения (TTC), и по мере приближения времени к TTC, штрафы за предупреждения и другие ложные тревоги усиливаются. w_{COC} предоставляет небольшую награду, когда предупреждение снято.

3. Моделирование и анализ результатов

A. Настройки моделирования

Параметры алгоритма установлены следующим образом. В соответствии с определением пространства состояний и пространства действий, алгоритм имеет 5 входов и 3 выхода. Архитектура нейронной сети как для сети актора, так и для сети критика состоит из двух скрытых слоев с 64 и 32 узлами, с коэффициентом обучения

6e-5 и порогом обрезки градиента 3.0.

Размер батча составляет 64, длина горизонта для исследования составляет 512, размер буфера для воспроизведения составляет 1e6, и сети обновляются многократно с использованием буфера воспроизведения, чтобы минимизировать потерю критика. Что касается формирования награды, коэффициент дисконтирования будущих наград (гамма) установлен на 0.975, а масштаб награды – 1. Процесс обучения прекращается, если общее количество шагов превышает 1 миллион.

На следующем рисунке 3 показан пример смоделированной среды столкновения самолетов, используемой для обучения и тестирования моделей предотвращения столкновений. Траектория слева представляет изменение высоты собственного самолета, а пунктирная линия справа представляет изменение высоты нарушителя. Начальная высота нарушителя – случайное значение в пределах 1000 футов относительно высоты собственного самолета. Для увеличения сложности экспериментального сценария, нарушитель летит по курсу столкновения с собственным самолетом в каждый момент времени. На траектории полета собственного самолета красные круги представляют собой предупреждения о снижении, выданные системой, а зеленые треугольники – предупреждения о наборе высоты, выданные самолетом. После выдачи предупреждения пилот выбирает, как отреагировать на предупреждение. Важно отметить, что горизонтальная ось не представляет относительное расстояние между двумя самолетами в горизонтальном направлении, а скорее TTC между двумя самолетами.

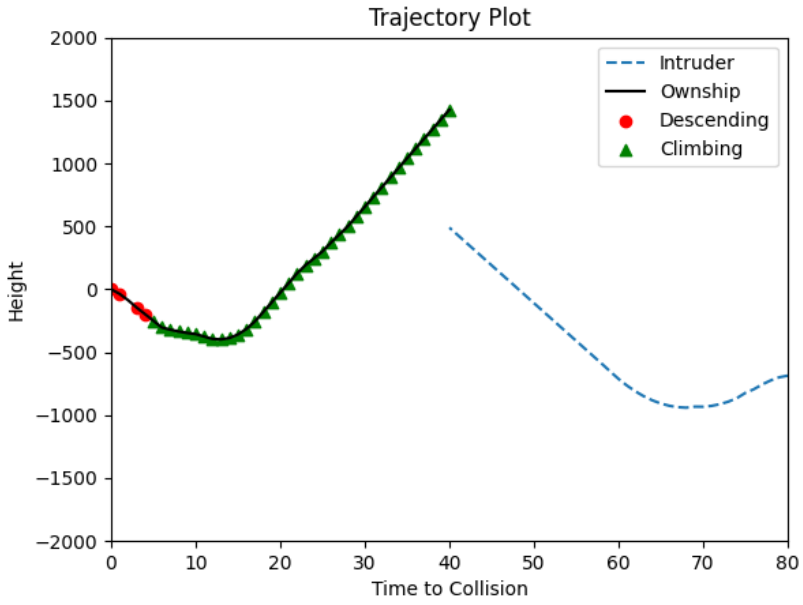

Рис. 3 Пример столкновения самолётов

B. Экспериментальные результаты и анализ процесса обучения

На основе вышеописанной среды моделирования и настройки параметров была обучена модель предотвращения столкновений с учетом реакции пилота на основе алгоритма DQN в условиях столкновения. Операционная система – Windows 11, графический процессор – GeForce RTX 3050. Для количественной оценки эффективности алгоритма в решении проблемы предотвращения столкновений мы использовали следующие метрики оценки:

- **Результаты сходимости средней награды:** средняя награда за последние 100 эпизодов.
- **Результаты сходимости частоты столкновений:** средняя частота столкновений за последние 100 эпизодов.
- **Результаты сходимости коэффициента успеха координации высоты (ACSR):** средняя доля эпизодов за последние 100 эпизодов, в которых

относительная высота самолетов была больше 900 футов, но меньше 1500 футов.

- **Результаты сходимости среднего количества предупреждений:** среднее количество предупреждений, выданных за эпизод, за последние 100 эпизодов.
- **Результаты сходимости среднего количества усилений советов:** среднее количество усилений советов за эпизод за последние 100 эпизодов.
- **Результаты сходимости среднего количества отмен советов:** среднее количество отмен советов за эпизод за последние 100 эпизодов.
- **Результаты сходимости среднего количества пересечений высот:** среднее количество раз, когда высоты обоих самолетов пересекались за эпизод, за последние 100 эпизодов.

Результаты показателей производительности алгоритма представлены в Таблице V. Кривые возврата эпизодов, частота столкновений и ACSR в процессе обучения алгоритма показаны на Рисунке 4, 5, 6 соответственно.

Из эпизодического вознаграждения на Рисунке 4 видно, что агент быстро накапливает опыт и за короткое время достигает относительно хорошего уровня вознаграждения. В ходе обучения дисперсия эпизодических вознаграждений непрерывно уменьшалась, и вознаграждение достигло своего пика примерно на 900,000 шагов. Как показано на Рисунке 5, агент быстро набрал опыт в решении проблемы предотвращения столкновений и смог снизить уровень столкновений до относительно низкого уровня в течение нескольких эпизодов. С дальнейшим обучением агент почти полностью предотвращал столкновения. Рисунке 6 показывает, что способность агента координировать относительную высоту двух самолетов

постоянно увеличивалась, достигнув пика около 95% к концу обучения. Однако, как показано в Таблице 5, уровень успеха координации за последние 100 эпизодов составил всего 63%, что указывает на нестабильность способности агента к координации.

Из Таблицы 5 можно увидеть другие плюсы и минусы алгоритма в решении проблемы предотвращения столкновений. В задаче предотвращения столкновений на 40 секунд агент в среднем инициирует 6.5 предупреждений, которые продолжаются в течение определенного времени. Во время этих предупреждений агент склонен изменять содержание предупреждений, что может вызвать определенное беспокойство у пилота. Однако преимуществом является то, что в задаче предотвращения столкновений агент редко приводит к пересечению высот двух самолетов, что представляет значительную угрозу для безопасности полетов.

Рисунки 7 и 8 соответственно иллюстрируют случай столкновения и успешное предотвращение столкновения с почти полной координацией высоты во время обучения. На Рисунке 7 система предотвращения столкновений выдала в общей сложности четыре предупреждения, но информация о предупреждении не получила достаточного внимания со стороны пилота, что в конечном итоге привело к столкновению. На Рисунке 8 система выдала два предупреждения, и пилот в некоторой степени отреагировал на них, успешно избегая столкновения двух самолетов. Общей чертой поведения системы в обоих случаях является наличие изменения предупреждений во время задачи предотвращения столкновений, что неблагоприятно для принятия пилотом.

Таблица 5. Общие результаты обучения алгоритма DQN

Алгоритм	Средняя награда	Частота столкновений	Уровень успеха координации	Количество оповещений	Количество укреплений оповещений	Количество изменений оповещений	Количество пересечений высоты
DQN	-29.37	0.02	0.63	6.5	1.8	23.4	0.1

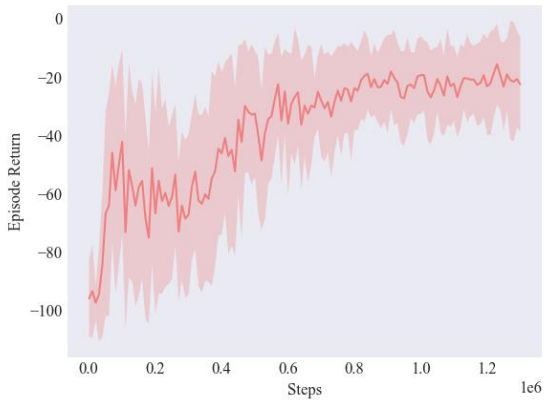

Рис. 4 Средний возврат за эпизод в процессе обучения

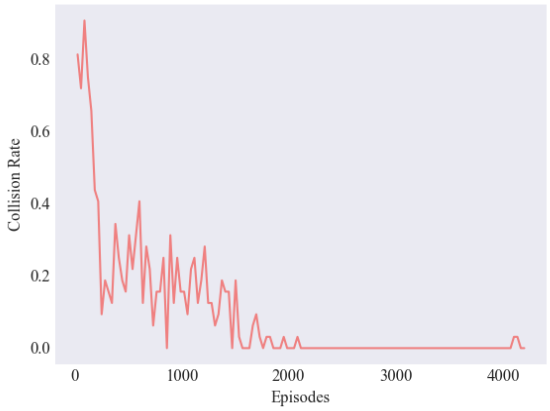

Рис. 5 Уровень столкновений в процессе обучения

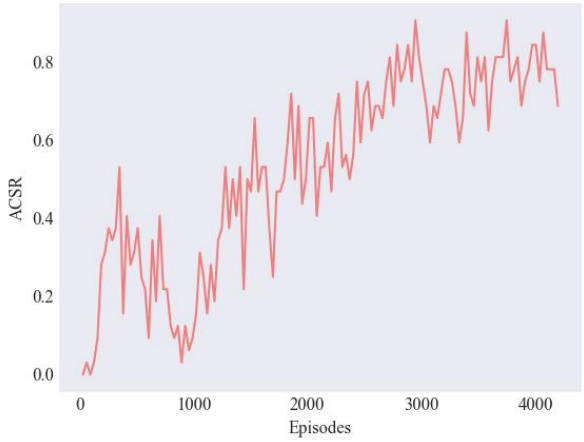

Рис. 6 Уровень успешной координации высот в процессе обучения

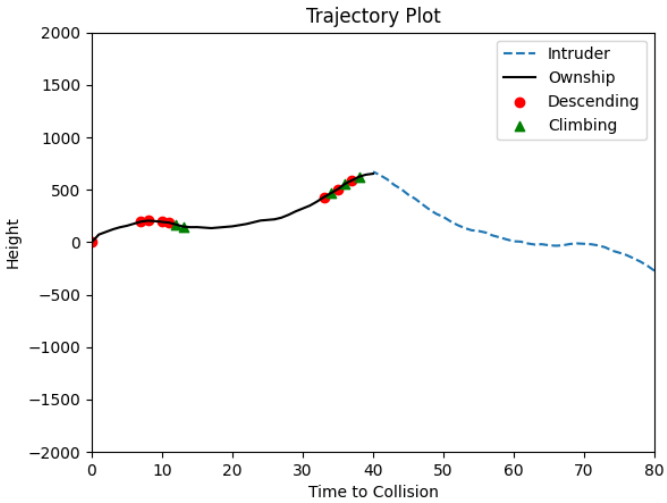

Рис. 7 Случай столкновения в процессе обучения

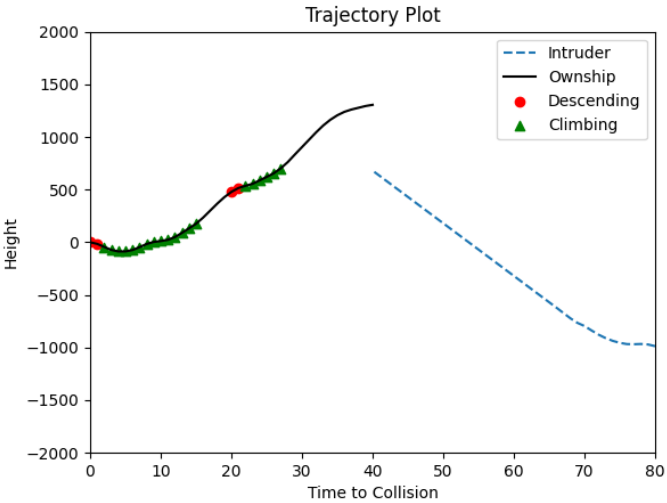

Рис. 8 Случай предотвращения столкновения и координации высоты в процессе обучения

C. Экспериментальные результаты и анализ процесса тестирования

В задаче предупреждения о столкновении существует три основных фактора, влияющих на эффективность предотвращения столкновений и координацию высот между двумя самолетами: максимальное отношение скоростей между самолетом-нарушителем и собственным самолетом, максимальное отношение ускорений и вероятность реакции пилота на предупреждения. Чем выше соотношения скоростей и ускорений нарушителя к собственному самолету и чем ниже приверженность пилота к предупреждениям, тем сложнее сценарий предотвращения столкновений, что затрудняет их предотвращение. В этом разделе мы используем те же базовые экспериментальные параметры, что и при обучении, изменяя соответственно максимальное соотношение скоростей, максимальное соотношение ускорений и вероятность реакции пилота на предупреждения, чтобы исследовать обобщаемость агента предотвращения столкновений, обученного на основе DQN и модели вероятности реакции пилота. Агент будет протестирован в каждом сценарии по 500 эпизодов, и его производительность будет проанализирована. Результаты тестов записаны в Таблицы 6, 7 и 8.

Из Таблицы 6 видно, что максимальная скорость собственного самолета всегда установлена на уровне 70 футов/с. По мере увеличения максимальной скорости нарушителя вероятность успешной координации высот двух самолетов быстро уменьшается. Когда максимальные скорости двух самолетов равны, вероятность координации до желаемой высоты составляет всего 3,6%, но в большинстве случаев все еще можно добиться успешного предотвращения столкновений, с вероятностью

столкновения всего 13,4%. Однако, когда максимальная скорость нарушителя увеличивается с 70 футов/с до 80 футов/с, вероятность неудачи предотвращения столкновения возрастает с 13,4% до 51,2%. При увеличении максимальной скорости нарушителя также увеличивается количество предупреждений, выдаваемых агентом, но другие показатели существенно не изменяются.

Таблица 7 показывает, что по мере увеличения максимального ускорения нарушителя вероятность столкновения остается на уровне 0, а вероятность успешной координации высот двух самолетов лишь незначительно уменьшается, что указывает на хорошую устойчивость обученного агента к изменениям максимального ускорения нарушителя. В течение этого процесса количество и качество предупреждений меняются незначительно. Таблица 8 иллюстрирует аналогичные изменения, где основным фактором изменения является вероятность того, что пилот предпримет действия, противоположные предупреждениям. По мере того как пилоты все чаще склонны действовать вопреки предупреждениям системы, успех избежания столкновений постепенно уменьшается. Однако даже при 40% вероятности противоположных действий вероятность успешного избежания столкновения остается на уровне 90%, хотя вероятность успешной координации высот двух самолетов составляет всего 24,4%. По мере увеличения вероятности нарушения пилотами предупреждений системы, система стремится уменьшить количество предупреждений, чтобы минимизировать беспокойство пилота, в то время как количество усиленных предупреждений в процессе предупреждения соответственно увеличивается.

Таблица 6.

Результаты тестирования при различных максимальных соотношениях скоростей между самолетами

Максимальная скорость интрудера	Частота столкновений	Уровень успеха координации	Количество оповещений	Количество укреплений оповещений	Количество изменений оповещений	Количество пересечений высоты
50/70	0.0	0.822	5.372	27.310	1.502	0.046
55/70	0.0	0.632	4.174	28.732	1.472	0.052
60/70	0.0	0.296	4.636	24.400	1.346	0.036
65/70	0.052	0.126	5.802	23.828	1.102	0.048
70/70	0.134	0.036	7.512	23.672	1.096	0.040
75/70	0.358	0.012	8.474	24.908	1.068	0.040
80/70	0.512	0.002	8.608	25.566	1.142	0.074

Таблица 7.

Результаты тестирования при различных максимальных соотношениях ускорений между самолетами

Максимальное ускорение интрудера	Частота столкновений	Уровень успеха координации	Количество оповещений	Количество укреплений оповещений	Количество изменений оповещений	Количество пересечений высоты
7	0.0	0.822	5.372	27.310	1.502	0.046
8	0.0	0.824	5.454	27.504	1.256	0.016
9	0.0	0.782	5.444	27.396	1.274	0.036
10	0.0	0.790	5.330	27.664	1.336	0.032
11	0.0	0.760	5.458	27.254	1.410	0.044
12	0.0	0.738	4.728	28.624	1.284	0.026
13	0.0	0.740	4.864	28.500	1.404	0.048

Таблица 8.

Результаты тестирования при различной вероятности ответа пилота

Вероятность реакции пилота	Частота столкновений	Уровень успеха координации	Количество оповещений	Количество укреплений оповещений	Количество изменений оповещений	Количество пересечений высоты
[0.5, 0.3, 0.2]	0.0	0.822	5.372	27.310	1.502	0.046
[0.5, 0.25, 0.25]	0.004	0.628	3.878	30.648	1.200	0.064
[0.45, 0.25, 0.3]	0.014	0.494	3.550	31.632	1.164	0.034
[0.4, 0.25, 0.35]	0.052	0.308	2.908	32.944	0.070	0.070
[0.4, 0.2, 0.4]	0.104	0.244	2.472	33.956	0.934	0.084

Заключение

Существующие системы предотвращения столкновений используют псевдокод или моделируют проблему предотвращения столкновений самолетов как MDP (Марковский процесс принятия решений) и применяют методы динамического программирования для определения оптимальной стратегии, что обеспечивает высокий уровень безопасности полетов. Однако слишком большие числовые таблицы, необходимые для хранения оптимальной стратегии, неточности, возникающие из-за интерполяционных методов, и вариативность реакций пилотов создают трудности для интеграции системы предотвращения столкновений с авионикой и ее дальнейшего развития.

В ответ на эти вызовы мы подошли к проблеме предотвращения столкновений как к задаче без модели, применяя методы глубокого обучения с подкреплением для решения проблемы NMAC (практически неизбежного столкновения) и стремясь к

более обоснованному подходу, который использует вероятностную модель для учета вариативности реакции пилота вблизи. В этой работе мы сначала представляем текущее состояние исследований по TCAS (система предупреждения столкновений в воздухе) и соответствующие теории глубокого обучения с подкреплением. Затем мы построили среду симуляции, специально предназначенную для проблем предотвращения столкновений самолетов, смоделировали проблему предотвращения столкновений как MDP, установили систему вознаграждений, изменяющуюся в зависимости от TTC (времени до столкновения), и создали вероятностную модель для обработки вариативности реакции пилота. Мы применили алгоритм DQN для обучения агента решать проблему предотвращения столкновений самолетов, одновременно учитывая координацию высоты обоих самолетов. В заключение, мы протестировали эффективность и устойчивость алгоритма в симуляциях с различными максимальными отношениями скоростей, максимальными отношениями ускорений нарушителя и собственника, а также вероятностями реакции пилота на предупреждения.

В наших экспериментах мы предполагали, что пилоты будут немедленно реагировать на каждое предупреждение и не учитывали с должным вниманием задержки в реакции пилотов или ситуации, когда пилоты могут не иметь возможности отреагировать из-за собственных ограничений. В будущем исследовании будет проведена оптимизация модели с учетом этих аспектов.

Список источников

1. Holland J E, Kochenderfer M J, Olson W A. Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance[J]. Air Traffic Control Quarterly, 2013, 21(3): 275-297.

2. De, D., and Sahu, P., “A Survey on Current and NextGeneration Aircraft Collision Avoidance System,” International Journal of Systems, Control and Communications, Vol. 9, No. 4, 2018, pp. 306–337.

3. Kochenderfer, M. J., Holland, J. E., and Chryssanthacopoulos, J. P.,“Next-Generation Airborne CollisionAvoidance System,” Lincoln Laboratory Journal, Vol. 19, No. 1, 2012, pp. 17–33.

4. Kochenderfer M J, Chryssanthacopoulos J P. Robust airborne collision avoidance through dynamic programming[J]. Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371, 2011, 130.

5. Kochenderfer M J, Amato C, Chowdhary G, et al. Optimized airborne collision avoidance[J]. 2015.

6. Julian K D, Kochenderfer M J, Owen M P. Deep neural network compression for aircraft collision avoidance systems[J]. Journal of Guidance, Control, and Dynamics, 2019, 42(3): 598-608.

7. Engel Y, Mannor S, Meir R. Reinforcement learning with Gaussian processes[C]//Proceedings of the 22nd international conference on Machine learning. 2005: 201-208.

8. Munos R, Moore A. Variable resolution discretization in optimal control[J]. Machine learning, 2002, 49: 291-323.

9. Julian K D, Sharma S, Jeannin J B, et al. Verifying aircraft collision avoidance neural networks through linear approximations of safe regions[J]. arXiv preprint arXiv:1903.00762, 2019.

10. Akintunde M, Lomuscio A, Maganti L, et al. Reachability analysis for neural agent-environment systems[C]//Sixteenth international conference on principles of knowledge representation and reasoning. 2018.

11. Ivanov R, Weimer J, Alur R, et al. Verisig: verifying safety properties of hybrid systems with neural network controllers[C]//Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. 2019: 169-178.

12. Kochenderfer M J, Espindle L P, Kuchar J K, et al. A comprehensive aircraft encounter model of the national airspace system[J]. Lincoln Laboratory Journal, 2008, 17(2): 41-53.

13. J. K. Kuchar and A. C. Drumm, “The Traffic Alert and Collision Avoidance System,” Lincoln Laboratory Journal, vol. 16, no. 2, pp. 277–296, 2007.

14. Collision Avoidance System Optimization with Probabilistic Pilot Response Models

15. Bertram J, Wei P, Zambreno J. A fast Markov decision process-based algorithm for collision avoidance in urban air mobility[J]. IEEE transactions on intelligent transportation systems, 2022, 23(9): 15420-15433.

16. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.

17. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.

18. Zhao Y, Liu P, Zhao W, et al. Twice sampling method in deep Q-network[J]. Acta Automatica Sinica, 2019, 14: 1870-1882.

19. Lin L J. Reinforcement learning for robots using neural networks[M]. Carnegie Mellon University, 1992.

20. Lim Y, Gardi A, Sabatini R, et al. Avionics human-machine interfaces and interactions for manned and unmanned aircraft[J]. Progress in Aerospace Sciences, 2018, 102: 1-46.

References

1. Holland J E, Kochenderfer M J, Olson W A. Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance[J]. Air Traffic Control Quarterly, 2013, 21(3): 275-297.

2. De, D., and Sahu, P., “A Survey on Current and NextGeneration Aircraft Collision Avoidance System,” International Journal of Systems, Control and Communications, Vol. 9, No. 4, 2018, pp. 306–337.

3. Kochenderfer, M. J., Holland, J. E., and Chryssanthacopoulos, J. P.,“Next-Generation Airborne CollisionAvoidance System,” Lincoln Laboratory Journal, Vol. 19, No. 1, 2012, pp. 17–33.

4. Kochenderfer M J, Chryssanthacopoulos J P. Robust airborne collision avoidance through dynamic programming[J]. Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371, 2011, 130.

5. Kochenderfer M J, Amato C, Chowdhary G, et al. Optimized airborne collision avoidance[J]. 2015.

6. Julian K D, Kochenderfer M J, Owen M P. Deep neural network compression for aircraft collision avoidance systems[J]. Journal of Guidance, Control, and Dynamics, 2019, 42(3): 598-608.

7. Engel Y, Mannor S, Meir R. Reinforcement learning with Gaussian processes[C]//Proceedings of the 22nd international conference on Machine learning. 2005: 201-208.

8. Munos R, Moore A. Variable resolution discretization in optimal control[J]. Machine learning, 2002, 49: 291-323.

9. Julian K D, Sharma S, Jeannin J B, et al. Verifying aircraft collision avoidance neural networks through linear approximations of safe regions[J]. arXiv preprint arXiv:1903.00762, 2019.

10. Akintunde M, Lomuscio A, Maganti L, et al. Reachability analysis for neural agent-environment systems[C]//Sixteenth international conference on principles of knowledge representation and reasoning. 2018.

11. Ivanov R, Weimer J, Alur R, et al. Verisig: verifying safety properties of hybrid systems with neural network controllers[C]//Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. 2019: 169-178.

12. Kochenderfer M J, Espindle L P, Kuchar J K, et al. A comprehensive aircraft encounter model of the national airspace system[J]. Lincoln Laboratory Journal, 2008, 17(2): 41-53.

13. J. K. Kuchar and A. C. Drumm, “The Traffic Alert and Collision Avoidance System,” Lincoln Laboratory Journal, vol. 16, no. 2, pp. 277–296, 2007.

14. Collision Avoidance System Optimization with Probabilistic Pilot Response Models

15. Bertram J, Wei P, Zambreno J. A fast Markov decision process-based algorithm for collision avoidance in urban air mobility[J]. IEEE transactions on intelligent transportation systems, 2022, 23(9): 15420-15433.

16. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.

17. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.

18. Zhao Y, Liu P, Zhao W, et al. Twice sampling method in deep Q-network[J]. Acta Automatica Sinica, 2019, 14: 1870-1882.

19. Lin L J. Reinforcement learning for robots using neural networks[M]. Carnegie Mellon University, 1992.

20. Lim Y, Gardi A, Sabatini R, et al. Avionics human-machine interfaces and interactions for manned and unmanned aircraft[J]. Progress in Aerospace Sciences, 2018, 102: 1-46.