NASA/TM–20240002174

Route-Recapturing State-Based Horizontal Maneuver Strategy for Automated Detect-and-Avoid

M. Gilbert Wu
Research Engineer, NASA Ames Research Center, Moffett Field, California, 94035-1000

February 2024

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English- language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at ***http://www.sti.nasa.gov***
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at 443-757-5803
- Phone the NASA STI Help Desk at 443-757-5802
- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–20240002174

Route-Recapturing State-Based Horizontal Maneuver Strategy for Automated Detect-and-Avoid

M. Gilbert Wu
Research Engineer, NASA Ames Research Center, Moffett Field, California, 94035-1000

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California, 94035-1000

February 2024

This report is available in electronic form at

https://ntrs.nasa.gov/

Abstract

This report describes a novel approach to the development of a horizontal maneuver guidance strategy for Detect-and-Avoid systems. The maneuver guidance strategy provides a directive turn action that can be automatically executed by the vehicle's auto-pilot system, taking into account the cost of recapturing the flight plan path. Pairwise conflict scenarios with non-accelerating intruders are simulated to validate the effectiveness of the maneuver guidance strategy. Initial results suggest the strategy is more effective for faster ownship than for slower ownship, which is unable to avoid conflict in certain scenarios against fast intruders. These findings indicate this novel approach shows great potential, but improvement to its performance is necessary and will be future work.

1 Introduction

Successful integration of Uncrewed Aircraft System (UAS) operations into airspaces populated with crewed aircraft relies on an effective Detect and Avoid (DAA) System. A DAA system provides surveillance, alerts, and maneuver guidance to keep a UAS "well clear" of other aircraft [1–3]. The look-ahead time of a DAA system for potential conflicts with other aircraft is typically between 1 and 3 minutes, providing an additional layer of safety on top of a collision avoidance system, which alerts for conflicts predicted to occur in less than 1 minute. Numerous DAA maneuver guidance algorithms have been developed for research [4, 5] as well as for operational use [6, 7]. These systems' DAA maneuver guidance follows the recommendations of RTCA SC-228's DO-365C, assuming a pilot in the loop and implementing "suggestive" maneuver guidance, which provides a range of feasible maneuvers (or a range of infeasible maneuvers) for the pilot to evaluate and act upon. A deep neural network approach has been proposed for providing DAA alerts and suggestive guidance [8].

Directive maneuver guidance of DAA systems, in comparison to suggestive maneuver guidance, can significantly improve pilots' ability to respond to DAA alerts [9]. Directive maneuver guidance provides a single maneuver in each dimension, e.g., one target ground course for horizontal maneuvers and one target altitude for vertical maneuvers. With a UAS pilot "in the loop", however, many pilots subjectively prefer self-derived maneuver options, using suggestive maneuver guidance as their basis [9]. On the other hand, directive maneuver guidance is suitable for DAA automation, which can support various use cases [10] such as failed datalink, pilots' inability to decide and initiate action, workload reduction, and alternative operational concepts in which pilots' role in DAA is replaced by automation. Several NASA research algorithms such as Independent Configurable Architecture for Reliable Operations of Unmanned Systems (ICAROUS) [11], and AutoResolver [12], provide directive maneuver guidance for DAA. Airborne Collision Avoidance System (ACAS)-Xu, designed for operational use by uncrewed aircraft, also provides directive maneuver guidance but only up to about 1 minute lookahead [13]. ICAROUS includes a software module that invokes Detect and AvoID Alerting Logic for Unmanned Systems (DAIDALUS) [5], receives suggestive guidance from it, and selects a particular maneuver according to a user-defined optimization goal and fitness function. This selected maneuver is checked against the mission-specific conformance criteria. AutoResolver was designed originally as conflict avoidance support tools from air traffic controllers' perspective, and was later extended and applied to DAA alerting and maneuver guidance [14]. AutoResolver takes user-defined preference of conflict-avoiding maneuver types, such as altitude change, speed change, or path stretches (turn-out followed by turn-back), as well as maneuver parameters, and performs a heuristic search for conflict-free solutions. The search stops after finding an acceptable solution (usually meaning conflict-free) without attempting to find an optimal solution. A deep reinforcement learning algorithm [15] generates maneuver guidance strategies that have similar turn-out and turn-back segments that connect the ownship from its current position to a route-recapture point on the ownship's flight plan.

This report presents a novel approach to the development of a directive, DAA horizontal maneuver guidance strategy in support of the concept of automated DAA. The development of this strategy is formulated as a reinforcement learning problem, in which the agent (the uncrewed aircraft in this case) learns the best conflict avoiding maneuver action by interacting with the environment. Rewards from the environment determines the "value" of an action, which is a function of the system's state. The values are estimated using local linear approximation [16] and solved via dynamic programming. The approach in this report is distinct from previous work in the following areas:

1. It employs a novel reinforcement learning approach that can be potentially more adaptable and scalable to support varying and everchanging airspace operations, when compared to conventional, heuristic approaches.

2. Its maneuver strategy is not limited to a list of coded conflict-resolving maneuver patterns such as a turn-out followed by a turn-back.

3. It avoids conflicts while taking into account the cost of recapturing the flight plan path.

4. It provides directive maneuver guidance which is naturally suitable for DAA automation.

These characteristics distinguish this work from previous work, which may meet a few of these statements but not all of them. The technical approach in this work is inspired by the development of the action look-up table for ACAS [6]. Nonetheless, this approach is distinct in its taking into account the reward of a route-recapture waypoint. The route-recapture waypoint is assumed to be provided by an external algorithm such as an auto-pilot system. Vehicle's performance parameters assume medium-sized, fixed-wing UAs with a mission speed in the range of 40 to 100 kts. The developed maneuver strategy is validated with pairwise conflict scenarios with non-accelerating intruders. Sensor errors or flight intent uncertainties are not considered in this initial assessment and will be considered in follow-on work.

This report is organized as follows: Section 2 describes the approach. Section 3 describes the conflict scenarios used for validation of the strategy. Section 4 presents the results and discusses potential areas for improvement. Section 5 summarizes the findings in this report and discusses potential future work.

2 Approach

This work considers horizontal maneuvers as a means of avoiding conflicts with another aircraft. It assumes no coordination of maneuvers. The other aircraft, or the intruder, will not attempt to avoid the conflict. The maneuver strategy selects a left turn, a right turn, or no turn. The Uncrewed Aircraft (UA) aims to remain "well clear" with the intruder, in which the "well clear" is defined by a minimum horizontal distance between aircraft. While resolving the conflict, the UA takes into account the next segment it needs to fly back towards its flight plan path in its current maneuver planning. It is assumed the UA's mission type or status allows it to fly directly to a waypoint further down the flight plan path after resolving the conflict. The UA does not have to fly through the segment that it misses due to the horizontal maneuvers. This assumption can be relaxed by slight modification to the formulation of the reinforcement learning problem to be described below.

Conflict-avoidance strategies formulated as a reinforcement learning problem have been studied by numerous research groups, many of which represent the strategies as neural networks. A recent literature review is provided here [17]. In the present work, the maneuver strategy is computed offline and stored as a look-up table for the "value" function. Grid points are defined within this seven-dimensional state space, and an approximate solution is iteratively obtained by assuming linearity of the value function between grid points, i.e., local linear approximation [16]. Alternative ways of estimating the value function, such as with neural networks, have been proposed [18].

Figure 2 shows the 7 state variables defining the system, which include the ownship, the intruder, and the route-recapture waypoint's positions and velocity, which in principle add up to 12 degrees of freedom in a 2-D plane. However, the route-recapture waypoint has zero velocity, which eliminates 2 degrees of freedom. The 2-D translational degeneracy and rotation degeneracy of the system further reduces the degrees of freedom by 2 and 1, respectively. Therefore, the system's state only has 7 variables. These state variables use relative position coordinates with respect to the ownship's state.

Only two ownship speeds, denoted as v_o, are considered in this work, 40 kts and 100 kts. These two speeds correspond roughly to the lower and upper bounds of the speed range assumed for low-size, weight, and power radar requirements defined in DO-366A, RTCA's Minimum Operational Performance Standards (MOPS) for radar in support of DAA systems. The intruder speed ranges from 0 kts to 170 kts, the assumed speed range for the majority of non-cooperative intruders defined by DO-365C (RTCA's DAA MOPS) and DO-366A.

Interaction with the environment produces several types of rewards after the ownship takes an action. At every non-terminal state (see below for terminal states' definition,) the ownship must take one of the following three options:

1. Left turn by 5° ground course, holding speed.

2. Hold its current ground course and current speed.

3. Right turn by 5° ground course, holding speed.

The time interval for action taking is 1 second. The action, denoted as a, transitions the system's state from s to s', where s' can have a probability distribution in general.

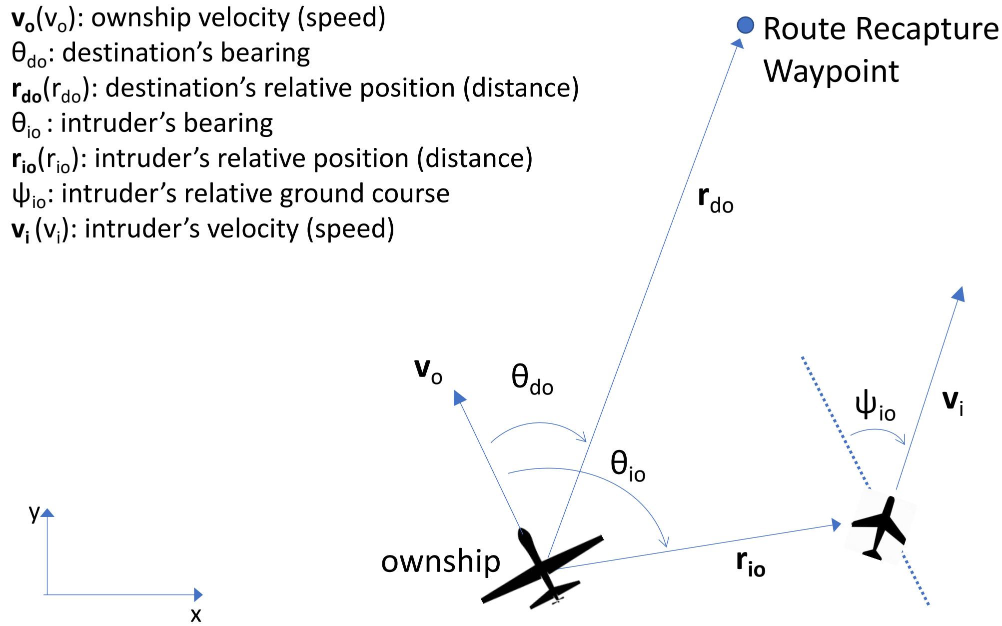

Figure 1. Local coordinates of the system

This work only considers a deterministic state transition, i.e., $s' = s'(s,\ a)$. The effect of wind is not taken into account, and it was assumed the speed can be held with respect to the ground.

The reward of taking an action a, R, provides feedback to the system and encourages the system to take actions that result in positive rewards. R is a function of the state s, the action a, and the resulting state s'. Since $s' = s'(s,\ a)$ for this study, s' can be dropped from the reward function's notation, i.e.,

$$R = R(s,\ a). \tag{1}$$

The following rewards are provided by the environment:

1. Positive reward when the ownship reaches the route-recapture waypoint, i.e., s' has $r_{do} < r_{do}^c$, where ${r_{do}}^c$ is the tolerance of the route-recapture waypoint
2. Negative reward when the ownship is within a distance from the intruder, r_{io}^0
3. Negative reward if the action involves a turn
4. Negative reward for the elapsed time.

The value function U is the accumulative effect of the reward function and is computed by this equation:

$$\begin{aligned} U(s) &= \text{argmax}_a \left[R(s,\ a) + U(s')\right], && \text{if } s \text{ is not a terminal state,} && \text{(2a)} \\ &= 0, && \text{if } s \text{ is a terminal state.} && \text{(2b)} \end{aligned}$$

The terminal states are states for which at least one of the following is true:

1. The ownship is within distance r_{do}^{c} from the route-recapture waypoint
2. The ownship's distance from the route-recapture waypoint exceeds a threshold, $r_{do}^{\max}$
3. The ownship's distance from the intruder exceeds a threshold, $r_{io}^{\max}$.

Once the system reaches a terminal state, no action is available to transition the system's state.

The following considerations are taken into account when selecting values of various rewards:

1. The reward of reaching the route-recapture point must be large enough to overcome time and turn penalty (negative rewards) a trajectory accumulates during a DAA look-ahead timespan.
2. The penalty of a severe loss of well clear, such as near-mid-air collision (NMAC), should be more than can be compensated for by the positive reward of reaching the route-recapture point. As such, the ownship should be discouraged from flying through the close neighborhood of the intruder in order to reach the route-recapture point.

Table 2 lists the the reward and terminal state parameters. Note that $r_{io}^{0} = 3300$ ft is greater than the well clear distance of 2,200 ft so as to create a protective "warning" buffer. ${r_{io}}^{\max}$ is the distance for which the two aircraft are well separated from each other and have a worst-scenario collision time of 120 seconds, well above the DAA look-ahead time.

Table 1. Reward and terminal state parameters

Parameter	Value
Recapture Waypoint (WP) Radius	500 ft
Reward for reaching the recapture WP	1.0
Reward for loss of well clear	$-C \times \left(1 - \sqrt{\frac{r_{io}}{r_{io}^{0}}}\right)$
C	1
r_{io}^{0}	3300 ft
Reward for flight time	-0.001
Reward for turning	-0.00001
r_{do} related terminal state	$r_{do} > 7$ NMI for $v_o = 40$ kts $r_{do} > 10$ NMI for $v_o = 100$ kts
${r_{io}}^{\max}$	56710 ft

An exact numerical solution to $U(s)$ is computationally challenging and is not attempted. Instead, $U(s)$ is approximated by piece-wise multi-linear functions. To do so, a 7-dimensional grid is constructed, and a dynamic programming technique called Gauss-Seidel method [16] is applied to iterate over $U(s)$ at grid points until $U(s)$ stabilizes. The selection of grid points is a trade-off between accuracy and computational demand. The grid points are denser in regions of the state space where $U(s)$ is expected to change rapidly, i.e., where the ownship is near the intruder, and less dense when $U(s)$ is expected to change slowly. For a specific ownship speed (40 kts or 100 kts), the total number of grid points is more than 40 million. It was found that, when the maximum change of $U(s)$ goes below 0.005 during the Gauss-Deidel iteration, the observed action policy in all validation scenarios becomes very stable. However, Convergence of $U(s)$ to 0.05 or below takes multiple days.

3 Conflict Scenarios

A large number of pairwise conflict scenarios with non-accelerating intruders are constructed by varying the following design parameters summarized in Table 3. These design parameters serve as input to the computation of the ownship and intruder's initial positions, initial velocities, and the route-recapture point's position. The route-recapture point is assumed to be some distance straight ahead of the ownship along its initial ground course. Since the system is translationally and rotationally invariant, the route-recapture point's position is arbitrarily set to the origin of the 2-D coordinate, and the ownship's initial ground course in arbitrarily set towards north, without loss of generality. Table 3 includes design parameters such as time to the closest point of approach (CPA), and horizontal miss distance (HMD), which is the predicted shortest distance between the two aircraft (at the time of the CPA).

Table 2. Conflict scenarios design parameters

Parameter	Values
v_o	40 or 100 kts
v_i	0 to 170 kts with a 5 kts interval
time to predicted CPA	60, 70, 80, and 90 seconds
time from predicted CPA to route-recapture	25, 30, 35, 40, 45, 50, 55, 60 seconds
HMD	0, 500, 1000, 2000 ft
Passing behind (intruder)	true, false (only if HMD > 0 and $v_i > 0$)

A small number of conflict scenarios are discarded as unrealistic. These scenarios include those that have the initial positions of the ownship and the intruder being too close (less than 4000 ft), the route-recapture point being swept through by the intruder's well clear radius for a prolonged period of time ($>$ 30 seconds), $v_i = 0$ and the intruder's position is too close to the route-recapture point ($<$ 4000 ft), etc. The total number of the remaining conflict scenarios is about 200,000.

Each conflict scenario is simulated by starting with the initial positions and velocities of the ownship and intruder, applying the horizontal maneuver recommended by the value function to the ownship, and flying the two aircraft by 1 second. This is repeated until the system advances to a terminal state. This can be broken down to the following steps:

1. A scenario is selected form simulation.
2. The positions and velocities of the ownship and intruder define the system's state.

3. The system's state is represented by the 7-variable coordiantes indicated in Figure 2.

4. The system's state feeds into the value-function look-up table, and a value of $U(s)$ for each of the ownship actions (left turn, no turn, right turn in this work) is calculated. The action with the highest $U(s)$ is returned to the simulation engine.

5. The simulation engine executes the action and progresses the system in time by 1 second. The ownship may turn left, turn right, or stay on its ground course. The intruder maintains its ground course. Both flights maintain their speeds.

6. Simulation ends if the ownship reachs its destination point after executing the action.

7. Otherwise, go back to step 2 and repeat this process.

4 Results

Section 4.1 show the aggreate metrics computed from all the scenarios. Section 4.2 presents four select sceanrios and discusses their simulation results.

4.1 Aggregate Metrics

The following aggregate metrics are computed from simulation results of the conflict scenarios:

1. Whether the ownship reaches the route-recapture point

2. Whether there is a loss of well clear

3. Minimum distance between the aircraft

4. Time Ratio (Time efficiency)

5. Whether the ownship passes in front of the intruder with insufficient buffer (undesirable).

Metrics are aggregated by speed pairs defined by the ownship and intruder's speeds. The total number of speed pairs is 35 (intruder speeds between 0 and 170 kts) x 2 (ownship speeds at 40 and 100 kts) = 70.

For the first aggregate metric, the ownship reaches the recapture waypoint in all conflict scenarios. This is the desired, perfect outcome.

Figure 2 shows the percentage of validation scenarios that result in a loss of well clear after following the maneuver guidance. The probability of a loss of well clear is zero for all the 100 kts ownship scenarios, while there is a non-zero percentage of a loss of well clear for the 40 kts ownship scenarios when $v_i > 75$ kts. These scenarios in which a slow ownship maneuvers against a fast intruder present a particular challenge for DAA systems because the ownship must start its maneuver at a greater distance from the intruder to be able to maintain separation. For example, for a pairwise,

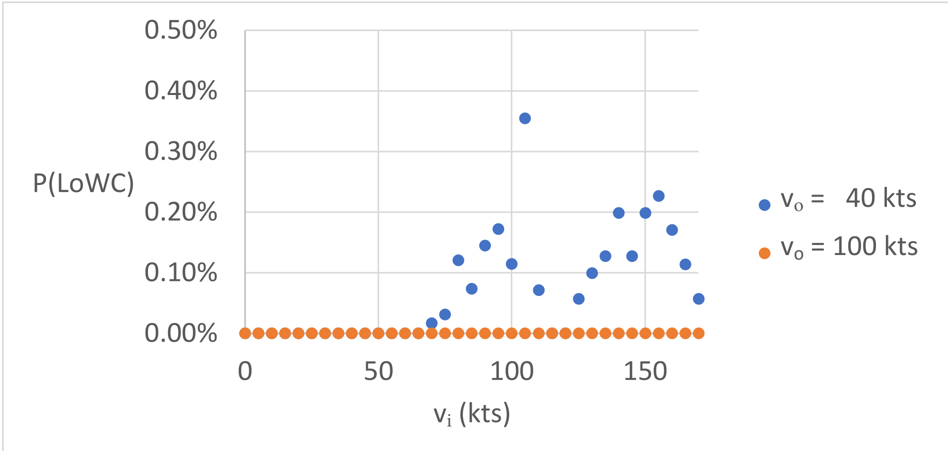

Figure 2. % of scenarios that lead to a loss of well clear

head-on encounter of a 40 kts ownship against a 170 kts intruder, RTCA DO-365C's Appendix D [19] indicates the ownship must start maneuvering before the aircraft's distance goes below 2.35 NM (maneuver initiation distance). Increasing the ownship speed to 100 kts drops the required maneuver initiation distance to 1.9 NM. On the other hand, decreasing the intruder speed to 100 kts also reduces the maneuver initiation distance to 1.5 NM. The ineffectiveness of the maneuver strategy for these speed pairs is likely due to the accumulation of errors arising from the multi-linear approximation of the solution to $U(s)$ over the particularly long maneuver initiation range. There appears to be a dip in P(LoWC) at v_{i} ∼120 kt. It is not clear whether this dip is statistically significant or the result of multiple interplaying factors.

Figure 3 shows the minimum distance between the ownship and the intruder among scenarios in each speed pair category. The minimum distance dips near 130 kts intruder speed to 600 ft for the 40 kts ownship results. This minimum distance gets close to the near-mid-air-collision distance of 500 ft as a result of ineffective maneuver guidance. This is definitely undesirable performance and must be improved with future versions of this algorithm.

Figure 4 shows the time ratio, denoted as TR, of the mitigated flight time following the maneuver strategy and the baseline flight time. Lower time ratios indicate better time efficiency. The baseline flight time is defined as the minimum flight time of a conflict-free ownship trajectory using a simple path stretch. The path stretch consists of an immediate turn-out segment and a turn-back segment, connected at an auxiliary point. This flight time is computed by a heuristic search in the space of the auxiliary point's position and is similar to the approach of AutoResolver's conflict-avoidance probe [12]. The time ratios for the 100 kts ownship are slightly lower than those of the 40 kts ownship scenarios, suggesting better time efficiency for the former.

The 5th and last metric is defined in this way: if the ownship follows the maneuver guidance and passes in front of the intruder, is the intruder's position within 5 seconds' flight time (using intruder speed) of violating the well clear? If yes, this is considered an undesirable "passing in front" event. No passing-in-front events with insufficient buffer are found in simulation results of all the conflict scenarios. This is also the

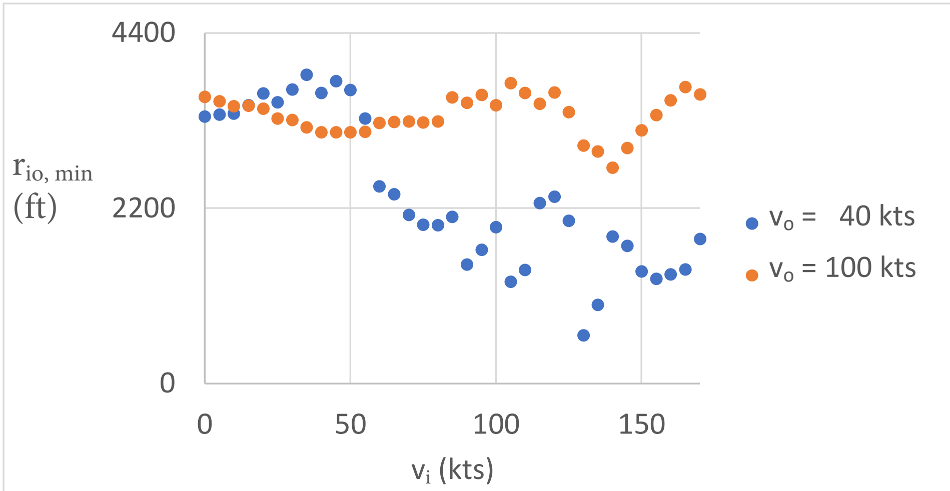

Figure 3. Minimum distance between the ownship and the intruder

desired, perfect outcome for this metric.

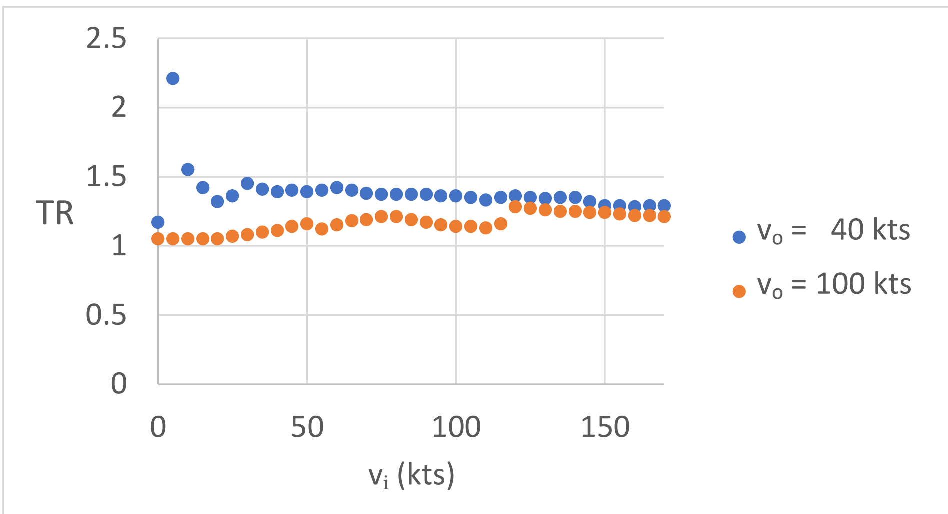

Figure 4. Average time ratio per speed pair

4.2 Select Scenarios

Representative mitigated trajectories from four scenarios are shown below. The intruder's trajectories are in red, while the ownship's is in blue. The small circle at $(0, 0)$ represents the route-recapture waypoint. The X symbol represents the starting point for the ownship and the intruder.

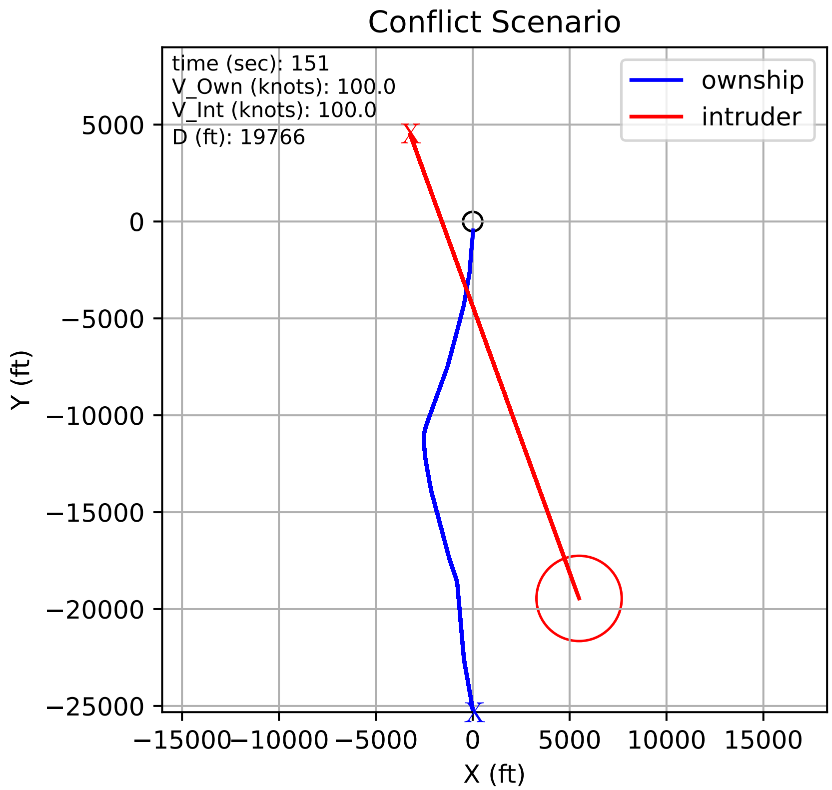

Figure 5. The ownship maneuvers behind the intruder

1. Simple, efficient maneuvers: Figure 5 shows results of a conflict scenario in which the ownship flies up and the intruder flies from top left to the bottom right. The ownship identifies a conflict with the intruder. A recapture waypoint is provided as input to the maneuver strategy algorithm, and is shown in this figure as the origin, $(0, 0)$. Figure 5 shows a successful and efficient ownship trajectory in which the ownship turns left, flies behind the intruder, turns back towards the recapture waypoint point, and reaches the recapture waypoint.

2. Wait-it-out maneuvers: Figure 6 shows a successful conflict-avoiding maneuver in which the ownship turns around to make a loop while the intruder passes. This extra circle is effective in "waiting it out" and avoiding the conflict. Note this type of maneuver strategy is automatically "discovered" by this

reinforcement-learning based algorithm. Heuristic algorithms such as AutoResolver or ICAROUS would not consider such a strategy unless it is specifically coded into their solution search logic.

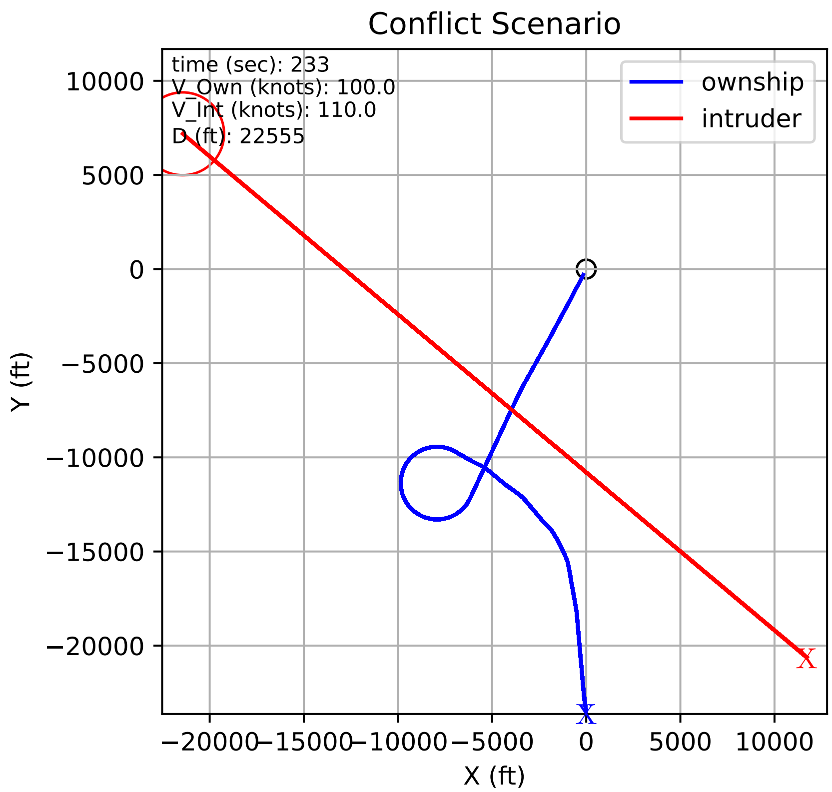

Figure 6. The ownship makes a loop to avoid a conflict with the intruder

3. Loss of well clear: Figure 7 shows the ownship being overtaken by an intruder and does not escape from the intruder's path in time. As Section 4.1 discussed, this occurs to a small fraction of conflict scenarios. It is undesirable performance and should be improved with future versions of this tool.

   One drawback of the current representation of the conflict in the form of a reward is it does not have the exponential decay behavior exhibited by a positive reward, and is confined to a single state where the negative reward is located [20]. This likely impacts the negative reward's effect to "deter" the ownship from approaching. Alternative representation of the negative reward arising from a conflict, called a "risk well", was proposed and appeared to lead to improved performance [20]. This alternative representation will be considered for future work.

4. Multiple loops near the route-recapture point: Figure 8 shows an interesting behavior. When avoiding a slowly moving intruder flying at 5 kts near the

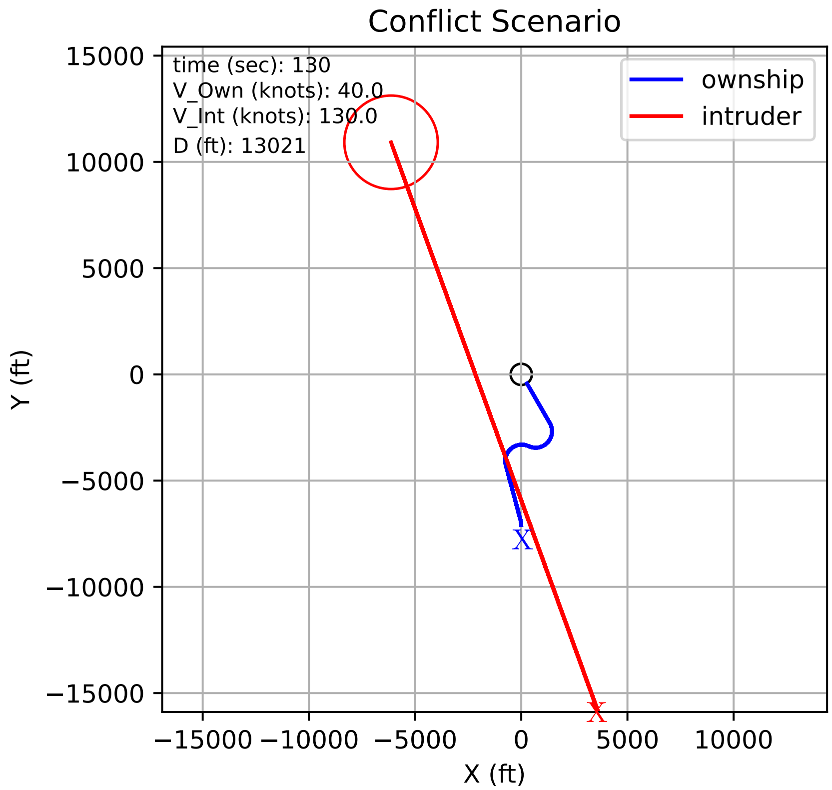

Figure 7. The ownship is overtaken by the intruder and leads to a loss of well clear.

recapture waypoint, the ownship misses the recapture waypoint on its first attempt and continues to make loops near the recapture waypoint. While this scenario exposes inefficiency in the maneuver strategy, this part of the state space in which the recapture waypoint is close to the intruder may be less important in operation since the it would have been a poor selection of the recapture waypoint by an external algorithm.

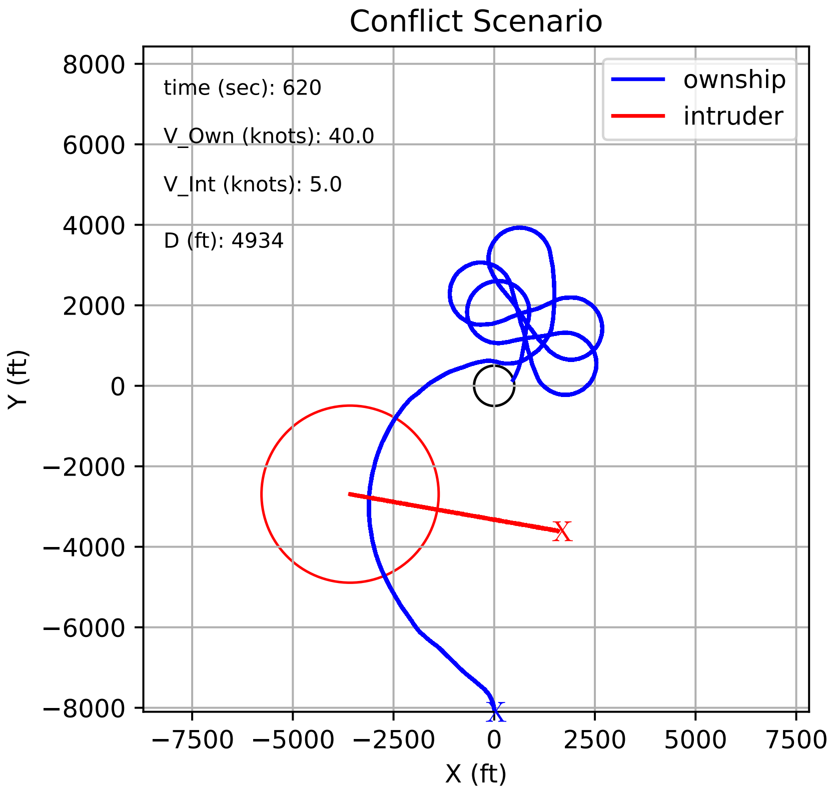

Figure 8. The ownship makes multiple loops near the route-recapture waypoint

5 Summary and Future Work

A novel approach to the DAA horizontal maneuver strategy for automated DAA systems has been proposed, formulating the conflict avoidance objective and route-recapturing objective as rewards in the context of reinforcement learning. Minimum-cost maneuver policy for this reinforcement problem is computed offline in an approximate way, using dynamic programming techniques, for aircraft performance parameter space representing medium-speed UAS and non-cooperative aircraft. The resulting strategy is a function of 7 state variables represented by a 7-dimensional grid of values.

Initial evaluation of this prepared maneuver strategy with pairwise, non-accelerating, conflict scenarios shows that, in all scenarios, the ownship successfully reaches the recapture waypoint by following the maneuver guidance. Inefficient maneuvers appear in a small number of scenarios in which the intruder moves slowly near the recapture waypoint. In terms of maintaining separation, 100% success rate is observed among scenarios involving the 100 kts ownship. The 40 kts ownship is able to maintain separation with intruders in a majority of scenarios. A small number of scenarios with the intruder speed between 75 kts and 170 kts leads to a loss of separation. Compared to research algorithms such as ICAROUS and AutoResolver, the current performance of this approach in terms of maintaining separation certainly needs improvement. Nonetheless, this approach, like a lot of machine-learning approaches, automaticaly discovers new types of conflict-avoiding maneuvers such as "wait-it-out", which is not achievable by heuristic algorihtms such as ICAROUS and AutoResolver unless this type of maneuvers is explicitly coded into their solution search logic.

Research on this approach shows promising results but also identifies areas for improvement. The following topics will be further pursued in the near term:

1. Improved performance of the separation metrics,

2. Ways of reducing computational time for the utility function,

3. Alternative way of representing intruder's negative reward [20].

Once these are explored, it will be interesting to compare the performance of this apporach to other research algorihtms such as ICAROUS and AutoResolver with a large number of conflict scenarios.

References

1. Filippone, E.; Corraro, F.; Corraro, G.; Ciniglio, U.; Shaw, C.; Pastor, E.; and Peinecke, N.: Definition of a Remain Well Clear Volume for IFR-RPAS Integration in Airspace Classes D-G in Europe. *AIAA Scitech 2023 Forum*, 2023. AIAA-2023-1590.

2. Cook, S. P.; Brooks, D.; Cole, R.; Hackenberg, D.; and Raska, V.: Defining Well Clear for Unmanned Aircraft Systems. *Proceedings of AIAA Infotech@ Aerospace*, AIAA-2015-0481, AIAA, 2015.

3. Johnson, M.; Mueller, E. R.; and Santiago, C.: Characteristics of a Well Clear Definition and Alerting Criteria for Encounters between UAS and Manned Aircraft in Class E Airspace. *Eleventh UAS/Europe Air Traffic Management Research and Development Seminar*, 2015, pp. 23–26.

4. Abramson, M.; Refai, M.; and Santiago, C.: The Generic Resolution Advisor and Confict Evaluator (GRACE) for Unmanned Aircraft Detect-And-Avoid Systems. NASA/TM-2017-219507, NASA Ames Research Center, 2017.

5. Muñoz, C.; Narkawicz, A.; Hagen, G.; Upchurch, J.; Dutle, A.; Consiglio, M.; and Chamberlain, J.: DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems. *34th Digital Avionics Systems Conference (DASC)*, IEEE/AIAA, 2015, pp. 5A1–1.

6. Owen, M. P.; Panken, A.; Moss, R.; Alvarez, L.; and Leeper, C.: ACAS Xu: Integrated collision avoidance and detect and avoid capability for UAS. *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, IEEE, 2019, pp. 1–10.

7. Suarez, B.; Kirk, K.; and Theunissen, E.: Development, Integration and Testing of a Stand-Alone CDTI with Conflict Probing Support. *Infotech@ Aerospace 2012*, 2012, p. 2487. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2012-2487`.

8. Ryu, J. Y.; Lee, H.; and Lee, H.-T.: Detect and Avoid AI System Model Using a Deep Neural Network. *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, IEEE, 2022, pp. 1–8.

9. Rorie, R. C.; and Fern, L.: The impact of Integrated Maneuver Guidance Information on UAS pilots Performing the Detect and Avoid Task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications Sage CA: Los Angeles, CA, vol. 59, 2015, pp. 55–59.

10. Theunissen, E.; and Veerman, T. H.: Automated Detect and Avoid: Autonomy and Ethics. *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*, IEEE, 2018, pp. 2A1–1.

11. Consiglio, M.; Muñoz, C.; Hagen, G.; Narkawicz, A.; and Balachandran, S.: ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, 2016, pp. 1–5.

12. Erzberger, H.; Lauderdale, T.; and Chu, Y.: Automated Conflict Resolution, Arrival Management, and Weather Avoidance for Air Traffic Management. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 226, no. 8, 2012, pp. 930–949.

13. Davies, J. T.; and Wu, M. G.: Comparative analysis of ACAS-Xu and DAIDALUS Detect-and-Avoid Systems. NASA Ames Research Center, 2018. NASA-TM-219973.

14. Gong, C.; Wu, M. G.; and Santiago, C.: UAS Integration in the NAS Project: Flight Test 3 Data Analysis of JADEM-Autoresolver Detect and Avoid System. NASA/TM-2014-218155, NASA Ames Research Center, 2016.

15. Pham, D.-T.; Tran, P. N.; Alam, S.; Duong, V.; and Delahaye, D.: Deep Reinforcement Learning Based Path Stretch Vector Resolution in Dense Traffic with Uncertainties. *Transportation Research Part C: Emerging Technologies*, vol. 135, 2022, p. 103463.

16. Kochenderfer, M. J.: *Decision Making under Uncertainty: Theory and Application*. MIT press, 2015. Chapter 4.

17. Wang, Z.; Pan, W.; Li, H.; Wang, X.; and Zuo, Q.: Review of Deep Reinforcement Learning Approaches for Conflict Resolution in Air Traffic Control. *Aerospace*, vol. 9, no. 6, 2022, p. 294.

18. Li, S.; Egorov, M.; and Kochenderfer, M.: Optimizing Collision Avoidance in Dense Airspace using Deep Reinforcement Learning. *Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)*.

19. *Minimum Operational Performance Standards (MOPS) for Detect and Avoid (DAA) Systems*. DO-365C, RTCA. Inc., 2022. See Figure D-13 for the maneuver initiation ranges for ownship of 40 kts.

20. Bertram, J.; Wei, P.; and Zambreno, J.: A Fast Markov Decision Process-Based Algorithm for Collision Avoidance in Urban Air Mobility. *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, 2022, pp. 15420–15433.