Collision Avoidance for Urban Air Mobility Vehicles using Markov Decision Processes

Project Category: Physical Sciences

Sydney M. Katz
Department of Aeronautics and Astronautics, *Stanford University*, Stanford, CA 94305
SUNet ID: smkatz

***Abstract*—Aircraft collision avoidance systems have long been a key factor in keeping our airspace safe. As Urban Air Mobility (UAM) systems are developed, it is important to create and evaluate new collision avoidance systems that are designed specifically for UAM vehicles. In this work, recent research in aircraft collision avoidance that formulates the problem as a partially observable Markov decision process (POMDP) is tailored to the expected behavior of a UAM vehicle at low altitudes. An optimal policy is generated for multiple reward structures and is evaluated through visualization and simulation. Ultimately, results show that with additional tuning, the methods in this work are an effective approach to UAM collision avoidance.**

I. Introduction

As Urban Air Mobility (UAM) systems edge closer to becoming reality, it is important to develop methods that ensure their safe integration into the airspace. A key milestone in this integration will be the development of robust collision avoidance systems for UAM vehicles. Aircraft collision avoidance systems represent the final layer of protection for an aircraft in a close encounter with another vehicle and therefore must be able to issue quick and accurate advisories. Therefore, developing a system that reacts safely across a wide range of scenarios requires sophisticated techniques in sequential decision making.

This work extends previous methods [1], [2] to develop a collision avoidance system for UAM aircraft operating at low altitudes, formulating the collision avoidance problem as a Markov decision process (MDP). The resulting policy is analyzed by plotting various slices of the state space. Additionally, the algorithm's performance is evaluated through Monte Carlo simulations of simple pairwise aircraft encounters. Based on these simulations, it is shown that the collision avoidance system is effective in improving vehicle safety in close encounters with other aircraft.

II. Related Work

The need for regulation of the airspace was revealed by a mid-air collision over the Grand Canyon in 1956, which ultimately led to the development of the Federal Aviation Administration (FAA) in 1958 [3]. Even with the increased regulation provided by the FAA, however, another mid-air collision occurred in 1978, this time over San Diego. It was this event that pushed the FAA to solidify the development of the Traffic Alert and Collision Avoidance System (TCAS) [4].

Since the 1980s, TCAS has been the primary collision avoidance logic mandated by the FAA; however, its complex structure and deterministic nature make it difficult to adapt to NextGen airspace procedures [1]. For this reason, the FAA has supported the development of a new system called the Airborne Collision Avoidance System X, or ACAS X. While TCAS relies on complicated heuristic rules to select advisories, ACAS X models the collision avoidance problem as a partially observable Markov decision process (POMDP) [1]. The state space is discretized, and the POMDP is solved using offline methods. The output is a lookup table of action costs for each state that can then be loaded onto the aircraft to provide alerts during flight. By utilizing techniques from artificial intelligence, ACAS X provides a more robust collision avoidance system that will be able to handle NextGen airspace procedures [2].

The broad structure of the POMDP formulation allows for easy adaptability of the algorithm to new aircraft types and operational environments. In addition to the version developed for manned aircraft, which became an international standard in 2018, versions have also been optimized for other aircraft types and environments [5]–[7]. The versions primarily differ in their operational considerations and the types of avoidance maneuvers (climb, descend, turn left or right, etc.) they command the aircraft to execute. For example, the system for manned aircraft, ACAS Xa, issues entirely vertical maneuvers [1], while the system for unmanned aircraft, ACAS Xu, has the option of turning left or right [5]. Another version proposed for quadcopters takes advantage of their unique ability to hover to allow the aircraft to make speed changes in order to avoid collisions [7]. This work develops and tests a collision avoidance system that meets the needs of UAM aircraft operating at low altitudes.

III. Problem Formulation

Consistent with the formulation of ACAS X, the UAM collision avoidance problem is modeled as a Markov decision process (MDP) with a discrete state and action space. This work assumes full observability, but the output could easily be extended to the partially observable case using the QMDP method [8]. An MDP is defined by the tuple (S, A, T, R, γ) where S defines the state space, A specifies the space of possible actions, T encodes the probabilistic transition model, R defines the reward function, and γ is the discount factor. The following subsections outline the design choices for each element, which are roughly based on those outlined in previous work [8], [9]. When referencing the individual aircraft in a pairwise encounter, the term ”ownship” refers to the UAM vehicle equipped with the collision avoidance system and the term ”intruder” refers to any aircraft the ownship may come across.

A. State Space (S)

There are two main design considerations when selecting the state space: the variables used to define the state and their discretizations. The state variables are based on those used in ACAS Xa [8]. These variables encode both the current relative position of the ownship and intruder aircraft as well as information regarding how the encounter will evolve such as their velocities and time to collision. The variables and their descriptions are summarized in table I. Adding the previous action to the current state allows the reward function to take into account past information without violating the Markov property. The horizontal state, such as the ground speeds and headings of both aircraft, is compactly represented by the variable τ.

TABLE I
STATE VARIABLES

State Variable	Description
h	relative altitude
$\dot{h}_0$	ownship vertical rate
$\dot{x}$	intruder vertical rate
a_{prev}	previous action
τ	time to horizontal closest point of approach

In addition to choosing the variables that represent the state, the range of values for each state variable must also be discretized. Values are chosen based on the anticipated behavior of UAM vehicles at low altitudes. The spacing between values is smaller in regions with high collision risk where taking an optimal action is critical. The discretizations are summarized in table II. The range for each variable was scaled significantly from the manned aircraft version to match the operational expectations of a UAM aircraft flying at low altitudes. The maximum value of τ, for example, was significantly increased. The range of τ should be chosen such that it allows sufficient time to resolve a conflict by issuing a maneuver. Because UAM vehicles will not be able to achieve the same vertical rates to avoid collisions as manned aircraft, they will likely take longer to climb out of a conflict situation. While previous work used a 40 second maximum [8], [9], this work uses 100 seconds.

TABLE II
STATE VARIABLES DISCRETIZATION

Variable	Units	Range (low:high)	Number of Values
h	ft	-600:600	39
$\dot{h}_0$	ft/min	-500:500	41
$\dot{h}_1$	ft/min	-500:500	41
a_{prev}	N/A	N/A	5
τ	s	0:100	101

B. Action Space (A)

The actions for the MDP are the maneuvers (called Resolution Advisories, or RAs) that the collision avoidance system tells the vehicle to execute. Because this collision avoidance system is meant to operate at low altitudes where obstacles such as buildings and terrain may be present, it is desirable to minimize lateral deviation from the flight path. For this reason, collision avoidance maneuvers in this system are purely vertical; however, since the vehicles are flying near the ground, descents are inhibited. The format of the actions is again similar to that of ACAS Xa [8], and they are summarized in table III. The minimum and maximum values indicate the vertical rate ranges considered to be compliant with the issued advisory.

TABLE III
ACTION SPACE

Action	Description	Min (ft/min)	Max (ft/min)
COC	clear of conflict	$-\infty$	∞
DNC	do not climb	$-\infty$	0
DND	do not descend	0	∞
$CL250$	climb	250	∞
$SCL450$	strong climb	450	∞

All actions may be commanded at any time except $SCL450$. In order to command this action, a climbing RA must already have been issued. In others words, $SCL450$ can only be commanded when the previous action was $CL250$ or $SCL450$. Any action that involves climbing or inhibits descending is said to have an upward sense, while any action that involves descending or inhibits climbing is said to have a downward sense. The only action with the downward sense is DNC. This presents a unique challenge for this system that was not present in previously developed vertical collision avoidance systems.

C. Transition Model (T)

The next state can be estimated using simple kinematics based on the current state and the likely actions of each aircraft. It is assumed that the ownship responds to its advisories with a noisy vertical acceleration and that the intruder continues on its current path with a noisy vertical acceleration centered at zero. This noise can be approximated in the transitions using sigma-point sampling as in previous work [7], [9]. In particular, each aircraft has a probability of taking each of three discrete accelerations based on the current state and action for nine total scenarios. The probabilities and accelerations for the ownship are summarized in table IV.

The acceleration assumed for the intruder is the same as the acceleration for the ownship with an advisory of COC. It

TABLE IV
TRANSITION PROBABILITIES

Action	Accelerations	Probabilities
COC	$-0.05g, 0, 0.05g$	$0.34, 0.33, 0.33$
DNC	$-0.1g, -0.15g, -0.2g$	$0.5, 0.3, 0.2$
DND	$0.1g, 0.15g, 0.2g$	$0.5, 0.3, 0.2$
$CL250$	$0.1g, 0.15g, 0.2g$	$0.5, 0.3, 0.2$
$SCL450$	$0.1g, 0.15g, 0.2g$	$0.5, 0.3, 0.2$

is important to note that these accelerations are only followed when an aircraft is not compliant with its current advisory (i.e. in the accepted vertical rate range specified by table II). If the current state indicates that the ownship is already following the advisory, no acceleration is necessary, and it is therefore set to zero.

The accelerations can be used to obtain a probability distribution over the next relative height and vertical rates. Updating the remaining state variables is trivial. The next a_{prev} is simply set to the current action, and the next τ is simply one second less than the current τ (assuming a 1 Hz update rate).

D. Reward Model (R)

Rewards must balance between safety and alert rate. Negative rewards can be thought of as penalties. Because the rewards significantly affect the performance of the algorithm, tuning the parameters is extremely important. For this reason, rather than using the full reward model used in ACAS X [8], this work is based loosely on a simpler reward function used in an initial proof of concept [9]. The parameters are summarized in table V.

TABLE V
REWARD PARAMETERS

Event	Reward
NMAC	-1
alert	-0.01
COC	0.0001
strengthening	-0.009
reversal	-0.1
invalid transition	-10

First and foremost, there is a large penalty for a near mid-air collision (NMAC). Traditionally, an NMAC is defined as a separation within 500 ft horizontally and 100 ft vertically. Since the MDP state defined in this work does not contain horizontal range, an NMAC is defined by any state in which $|h|$ is less than 100 ft and $\tau = 0$. In order to ensure that the collision avoidance system does not maximize safety by alerting all the time, alerts (defined as any action that is not COC) are penalized and the action of COC is given a small reward.

The final three rewards depend on the previous action. A strengthening occurs when the next action has the same vertical sense as the previous action but a larger commanded vertical rate. An example is transitioning from DND to $CL250$. A reversal constitutes a change in sense between the next and previous action. An example is transitioning from DNC to DND. The final reward enforces the permissible action transitions by associating a large penalty with an invalid transition.

IV. SOLUTION METHOD AND IMPLEMENTATION

With a discretized state and action space, the MDP can be solved using a form of dynamic programming called value iteration. Specifically, the Bellman equation is used to perform iterative updates to the value function $Q^*(s,a)$.

$$Q^*(s,a) = R(s,a) + \sum_{s' \in S} T(s,a,s') \max_{a' \in A} Q^*(s',a') \quad (1)$$

The MDP was implemented by modifying code in the following repository: https://github.com/sisl/VerticalCAS. The code relies on POMDPs.jl, a package written for the Julia language that provides a general framework for defining and solving POMDPs [10]. The modifications to the code for this project can be found at https://github.com/smkatz12/UAM_CAS.

Because of the nature of the transition for the state variable τ, it is possible to solve the MDP with a single pass through the state space using asynchronous value iteration. Specifically, because τ always decreases in the next state, the method begins at $\tau = 0$ and propagates the reward backwards. This method along with the highly parallelized implementation in the VerticalCAS repository allows the MDP to be solved in under 95 seconds on five Intel Core i7 processors operating at 4.20 GHz.

V. RESULTS

The resulting policy was analyzed in two ways. First, it was plotted over slices of the state space to develop an intuition behind the decisions it recommends. Second, its performance was evaluated on a test set of pairwise encounters.

In addition to the baseline reward structure presented in the preceding section, two other reward models were analyzed to better understand the tradeoff between safety and alert rate. In particular, in the first revision, an NMAC was redefined to include any state where $\tau < 5$ seconds and $|h| < 100$ ft. This better encodes the horizontal component of an NMAC. The final revision added an extra penalty for close proximity of the aircraft. This penalty takes effect when $\tau < 15$ seconds and $|h| < 125$ ft. The following sections demonstrate the results of these trades for both analysis methods.

A. Policy Analysis

The optimal policy can be visualized by plotting slices of the state space, specifically varying h and τ while holding the other state variables constant. Figure 1 and fig. 2 show the results for various ownship descent rates while the intruder remains in level flight. It is clear that the invalid transition penalty is working properly as the $SCL450$ RA is only issued in the plots where the previous RA was $CL250$.

A notable feature of the plots in fig. 1 is the gap in alerting between $h = 50$ ft and $h = 100$ ft , even for low values of τ. This can be explained by the fact that descents are inhibited.

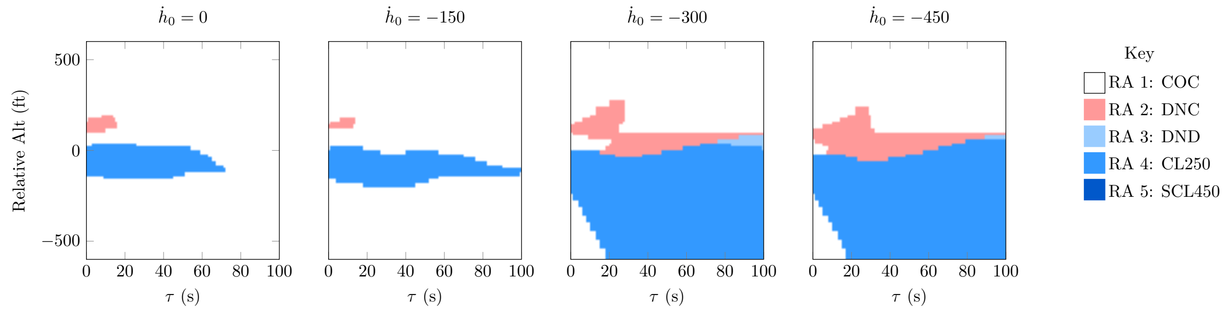

Fig. 1. Plots of optimal policy for various ownship vertical rates (in ft/min). The plots assume the intruder is flying level ($\dot{h}_1 = 0$) and that the previous RA is COC.

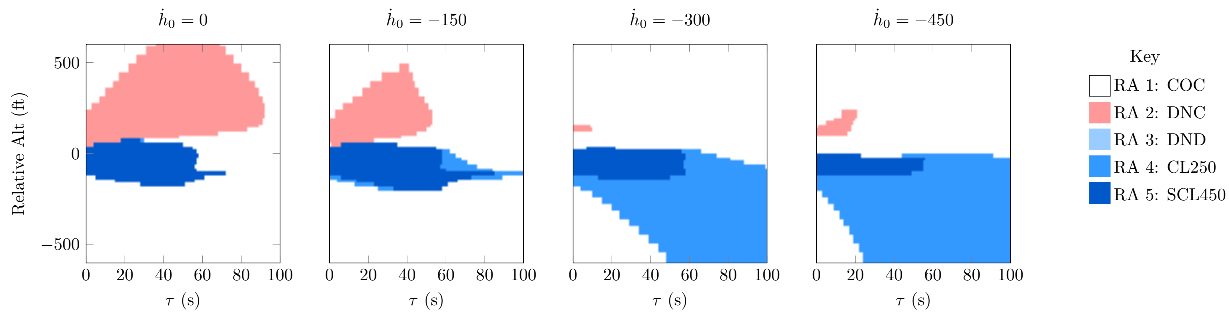

Fig. 2. Plots of optimal policy for various ownship vertical rates (in ft/min). The plots assume the intruder is flying level ($\dot{h}_1 = 0$) and that the previous RA is $CL250$.

When the intruder is flying in level flight 50-100 ft above the ownship, the most the system can do is tell the ownship not to climb; however, this will not resolve the conflict as both aircraft will remain in the range for an NMAC. Since an NMAC is unavoidable and alerts are penalized, the optimal policy is to refrain from alerting at these relative altitudes. Interestingly, this behavior is not observed in the range of $h = 0 - 50$ ft. In this region, the aircraft are close enough to one another that issuing a climb advisory provides the ownship with enough time to climb past the intruder's altitude.

Figure 3 show a comparison of the optimal policy from COC for the three reward structures tested. As the penalty for flying in close proximity increases, the size of the alert region also increases. This hints at a possible tradeoff between safety and alert rate that is confirmed by the simulation results shown in the next section.

B. Simulation Results

In order to test the safety and effectiveness of the collision avoidance system, a Monte Carlo simulation of 1,000 aircraft encounters between straight, level flying aircraft was created. Encounters were generated by sampling various trajectory features (e.g. airspeed, horizontal and vertical miss distances, heading, etc.) from pre-specified distributions and propagating the dynamics to obtain these features. The ownship follows its trajectory until it receives an alert, at which point it follows a normally distributed acceleration centered at $0.12g$ with a standard deviation of $0.03g$ until it reaches the compliant velocity range.

Figure 4 and fig. 5 show the horizontal and vertical profile of two example encounters. Both encounters show that the collision avoidance system is making sensible decisions to avoid collisions. Table VI summarizes the simulation results.

TABLE VI
SIMULATION RESULTS

Equipage	NMACs	Alerts	Induced	Resolved
unequipped	472	-	-	-
$\tau = 0$	217	611	0	255
$\tau < 5$	211	635	0	261
closeness	197	754	0	275

With no collision avoidance system, the encounter set results in 472 NMACs. Each policy tested reduces this number by over half. Additionally, the collision avoidance system did not induce any NMACs that were not present in the unequipped case.

As anticipated, more NMACs are resolved as the penalty for close proximity increases; however, the alert rate also increases significantly. This result illustrates the trade-off between safety and alert rate. This trade-off is further illustrated by fig. 6. In the first case, the system alerts too late and results in an NMAC, while the cases with a higher penalty alert soon enough to resolve the encounter.

VI. Conclusion

This paper has demonstrated a collision avoidance model for UAM vehicles flying at low altitudes that successfully resolves

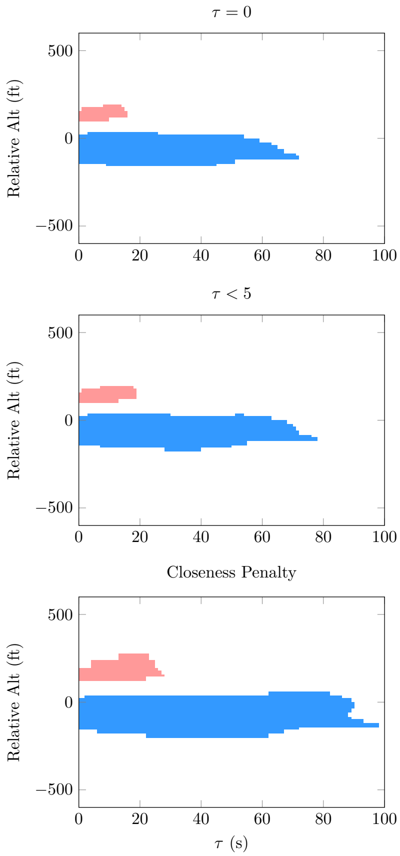

Fig. 3. Comparison of the optimal policy for various penalties on close proximity. Both aircraft are in level flight and the previous RA is COC

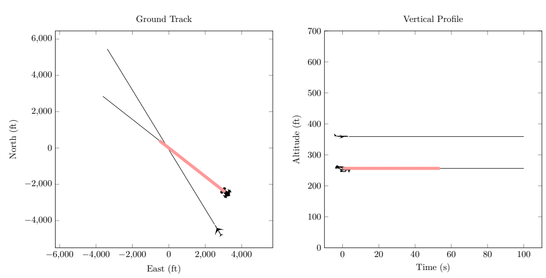

Fig. 4. Example simulated encounter with DNC alert. Alerts are represented by marks that correspond to the same key as the policy plots.

potential conflict scenarios. The model extends methods from prior work in order to formulate the collision avoidance problem as an MDP. The model was analyzed by examining the resulting policy at various portions of the state space as well as by simulating it on a set of 1,000 pairwise encounters. The policy and simulation results indicate a successful implementation.

This paper opens multiple paths for future work. Many of the unresolved NMACs in the encounter set are due to the gap in alerting identified in fig. 1. Future work should determine a method to address this issue, and further analysis of the policy plots should be conducted. Additionally, the output should be tested on more encounter sets that realistically encode UAM

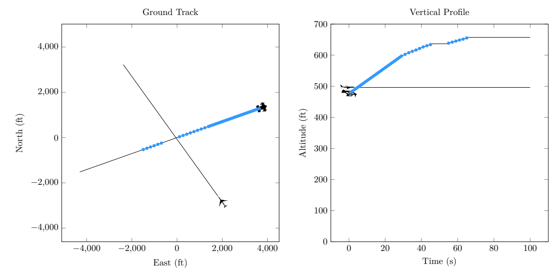

Fig. 5. Example simulated encounter with $CL250$ alert. Alerts are represented by marks that correspond to the same key as the policy plots.

vehicle behavior, and the reward function structure should be further tuned based on the results. Finally, the system should be tested on encounters in which both aircraft are equipped with a collision avoidance system to ensure that avoidance maneuvers are complementary.

VII. Acknowledgements

This work was done with the guidance of Professor Mykel Kochenderfer. The VerticalCAS repository that the code for this work was based on was developed by Kyle Julian. His helpful discussions throughout the progression of this work were appreciated.

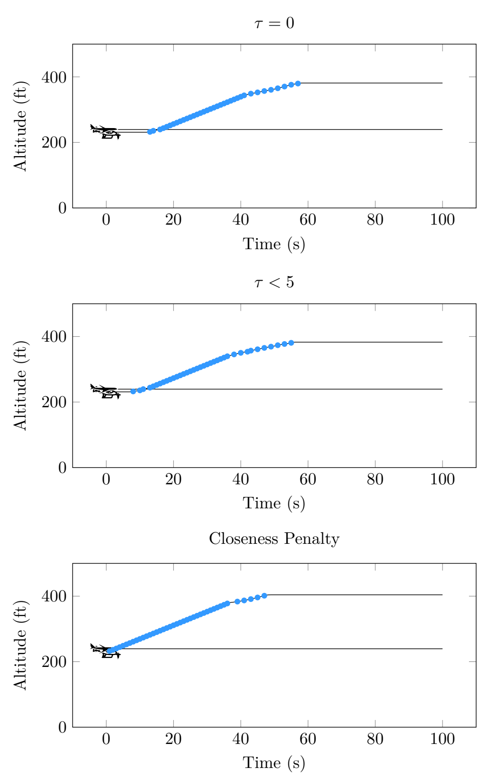

Fig. 6. Vertical profile of the same encounter with three different reward function structures corresponding to those in fig. 3. The top encounter results in an NMAC, while the other two are resolved.

REFERENCES

[1] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, "Next-generation airborne collision avoidance system," Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States, Tech. Rep., 2012.

[2] W. A. Olson, "Airborne collision avoidance system x," Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States, Tech. Rep., 2015.

[3] T. Williamson and N. A. Spencer, "Development and operation of the traffic alert and collision avoidance system (tcas)," *Proceedings of the IEEE*, vol. 77, no. 11, pp. 1735–1744, 1989.

[4] "Introduction to tcas ii," Federal Aviation Administration, Tech. Rep., 2011.

[5] M. Marston and G. Baca, "Acas-xu initial self-separation flight tests," 2015.

[6] K. A. Smith, M. J. Kochenderfer, W. A. Olson, and A. E. Vela, "Collision avoidance system optimization for closely spaced parallel operations through surrogate modeling," in *AIAA guidance, navigation, and control (GNC) conference*, 2013, p. 4624.

[7] E. R. Mueller, "Multi-rotor aircraft collision avoidance using partially observable Markov decision processes," Ph.D. dissertation, Stanford University, 2016. [Online]. Available: http://purl.stanford.edu/rv444dz2833

[8] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

[9] M. J. Kochenderfer and J. Chryssanthacopoulos, "Robust airborne collision avoidance through dynamic programming," *Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371*, 2011.

[10] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-300.html