Collision Avoidance for Unmanned Aircraft using Coordination Tables

Rachael E. Tompa,* Blake Wulfe,† Michael P. Owen,‡ and Mykel J. Kochenderfer*
*Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305
†Department of Computer Science, Stanford University, Stanford, CA, 94305
‡Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, 02420

***Abstract*—The safe integration of unmanned aircraft into the airspace requires reliable systems to keep aircraft separated. Unmanned aircraft will likely have to use horizontal maneuvers, in contrast with manned aircraft collision avoidance systems, which use only vertical maneuvers. Coordination of horizontal maneuvers is much less straightforward than vertical maneuvers because in some situations it is better for both aircraft to maneuver in the same direction and in other situations it is better to maneuver in opposite directions. This paper uses dynamic programming to generate a coordination scheme that is represented as a lookup table. Different aircraft that may have different collision avoidance systems can use this table to produce complementary maneuvers. The robustness of the table to various policies is measured in terms of safety and efficiency. The empirical results suggest that a coordination table can improve safety for encounters involving different collision avoidance systems.**

I. Introduction

The safe integration of unmanned aircraft into the airspace requires reliable systems to keep aircraft separated. In the context of manned aircraft, there have been several decades of research on collision avoidance [1]. The Traffic Alert and Collision Avoidance System (TCAS) provides vertical maneuver guidance to large manned aircraft [2]. It was found that coordinating maneuvers is important for ensuring safety, and so TCAS coordinates between aircraft so that the advisories issued to the aircraft are always in opposite directions [3]. In contrast to manned aircraft, unmanned aircraft will likely rely on horizontal maneuvers in many cases due to climb performance constraints and altitude restrictions.

The coordination of horizontal maneuvers is much less straightforward than vertical maneuvers because opposite directions are not always advised. For example, when aircraft are headed in the same direction, one should turn right and the other should turn left, but when they are head-on, they should either both turn left or both turn right. Determining the situations where aircraft should turn in the same direction or in opposite directions is not always obvious. It is hypothesized that safety can be enhanced by agreement between aircraft on a coordination scheme.

Currently, TCAS is mandated on all large manned aircraft [4]. Due to the wide variability of unmanned aircraft platforms, it is unlikely that all unmanned aircraft will be mandated to carry the same system. For example, there are several avoidance systems currently under development for large unmanned aircraft including ACAS Xu [5], JOCA [6], and MIDCAS [7] along with various systems that are targeted for smaller unmanned aircraft [8], [9].

A potential solution for coordination between different systems is to use a standard coordination table that all aircraft can agree upon and query during an encounter regardless of their policy for selecting a maneuver. This paper uses dynamic programming to create a variety of potential collision avoidance policies and a coordination table [10]. A collision avoidance policy is made using a pairwise method presented by [8] adapted for larger unmanned aircraft. The approach involves modeling an aircraft encounter as a Markov decision process (MDP) and solves for the optimal action for each aircraft state during an encounter. This approach was selected to align with the approach used to generate the Airborne Collision Avoidance System X (ACAS X), which is targeted as a replacement for TCAS [11].

To create a variety of policies, the model was adjusted by changing the weights on safety, efficiency, and separation. To generate the coordination table, the problem is modeled in a similar way but instead of keeping the actions for both aircraft, the coordination table stores whether both aircraft take actions in the same direction or opposite directions.

A simulation framework was created to evaluate the learned policies and coordination table. By evaluating the use of a coordination table on aircraft with heterogeneous policies we found that, with an appropriate intruder model, a coordination table can improve safety while maintaining alert rates. For encounters including a safe and adversarial policy the increase in safety is around 57%, and for encounters between two adversarial policies the increase in safety is around 25%.

The paper proceeds as follows. Section II describes how the aircraft encounter is modeled and how policies are generated. Section III explains how actions are selected during an encounter and how the coordination table is generated and used. Section IV describes the simulations run to test the success of a coordination table. Section V presents results. Section VI concludes and discusses areas of future work.

II. Problem Formulation

An MDP models a sequential problem with an agent that chooses an action at each time step based on the observed state [10], [12]–[14]. A transition model specifies the probability of transitioning to some new state conditioned on the current state and action. A reward model specifies the reward associated with different states and actions. An optimal policy associated with an MDP is one that maximizes the expected accumulation of reward over time. This section describes how the aircraft collision avoidance problem was modeled as an MDP and solved using dynamic programming to generate policies.

A. State and Action Space

To define the state space, one aircraft is denoted the master and the other, the slave. As with TCAS, the aircraft with the lower transponder address is the master. The state space of the problem is seven dimensional. The state variables capture the relative position of the slave, the relative heading of the slave, the speeds of both aircraft, and whether the individual aircraft are responding to their advisories. The master is located at the origin heading in the positive x-direction. The slave aircraft is located at range ρ and angle θ relative to the master as shown in Figure 1. The heading of the slave relative to the master is ψ. The individual aircraft speeds are v_{m} and v_{s} for the master and slave, respectively. We assume large unmanned aircraft (e.g, a Predator) with a cruise speed between 35 m/s and 45 m/s. Two Boolean state variables, r_{m} and r_{s}, represent whether the master and slave are responding to their advisories, respectively. The state space is discretized using a seven dimensional grid, resulting in a total of 1,577,089 states with the discretization shown in Table I. All of the state variables are discretized uniformly except for ρ, which has a finer discretization for smaller values.

The action space is a set of joint commands $(a_{\text{m}}, a_{\text{s}})$ that specify a bank angle or if the aircraft are not in conflict, no advisory. Each aircraft may receive one of six different bank angle commands: strong right (SR) commanding −12°, right (R) commanding −6°, straight (S) commanding 0°, left (L) commanding 6°, strong left (SL) commanding 12°, or no advisory (NA). Hence, there are 36 potential joint actions. The bank commands were selected based on the velocity range to achieve turn rates close to straight, a standard rate (3° per second) turn, and a half standard rate (1.5° per second) turn.

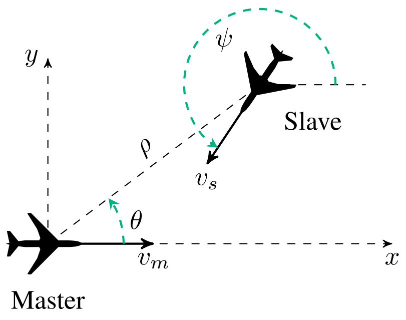


Fig. 1: State variables

TABLE I: Aircraft state variables

Variable	Minimum	Maximum	Values	Units
ρ	0	12,800	32	m
θ	$-\pi$	π	37	rad
ψ	0	2π	37	rad
v_{m}	35	45	3	m/s
v_{s}	35	45	3	m/s
r_{m}	0	1	2	NA
r_{s}	0	1	2	NA

B. Transition Model

The transition model specifies the distribution over the next state given the current state and joint action. When an aircraft is responding to an advisory, the banking response is modeled as a Gaussian distribution centered at the commanded bank angle with a standard deviation of 2°. When the action is NA or the aircraft is not responding, the bank angle is modeled as a Gaussian distribution centered at 0° with a standard deviation of 5°. The advisory bank angles are relatively small, and, therefore, the resulting bank angle changes occur instantly. The speed used to propagate the position is modeled as a Gaussian distribution centered at the given speed with a standard deviation of 4 m/s. The aircraft position is updated using Dubin's kinematic model [15].

C. Reward Model

The reward model is a combination of four negative rewards with values that can be adjusted to balance safety and efficiency. The rewards are additive and expressed as follows:

1) The *NMAC* reward penalizes near mid-air collisions (NMACs):

$$R_{\text{NMAC}} \times \mathbb{1}\{\rho < \rho_{\text{NMAC}}\} \tag{1}$$

where ρ_{NMAC} is the distance that defines an NMAC. As in prior studies, ρ_{NMAC} is set to 500 ft [16].

2) The *Close* reward penalizes proximity:

$$R_{\text{Close}} \times \exp\sqrt{\frac{-\rho}{\rho_{\text{NMAC}}}} \tag{2}$$

3) The *Maneuver* reward penalizes maneuvers:

$$R_{\text{maneuver}} \times \mathbb{1}\{a_{\text{own}} \neq \text{NA} \vee a_{\text{int}} \neq \text{NA}\} \tag{3}$$

4) The *Strength* reward penalizes maneuvers based on their strength:

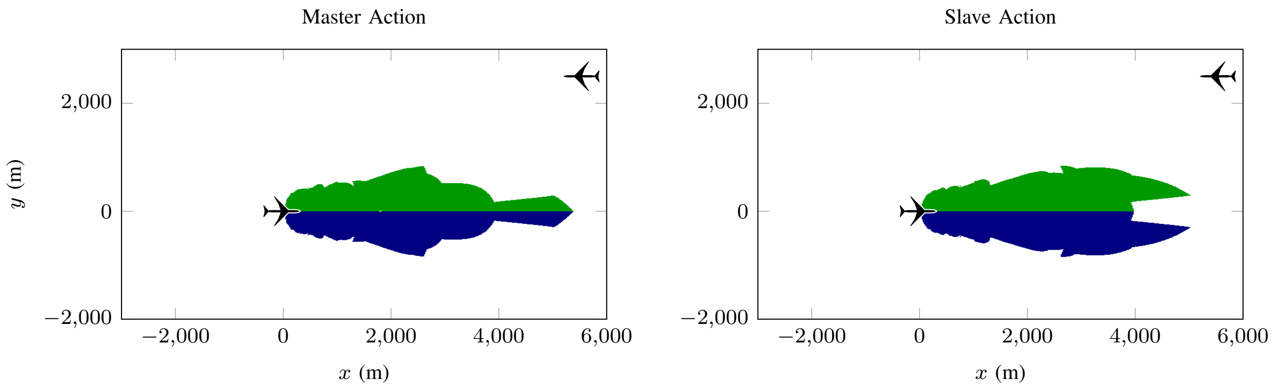
$$R_{\text{Strength}} \times (b(a_{\text{own}})^2 + b(a_{\text{int}})^2) \tag{4}$$

where b is a function that returns the bank angle of the advisory.

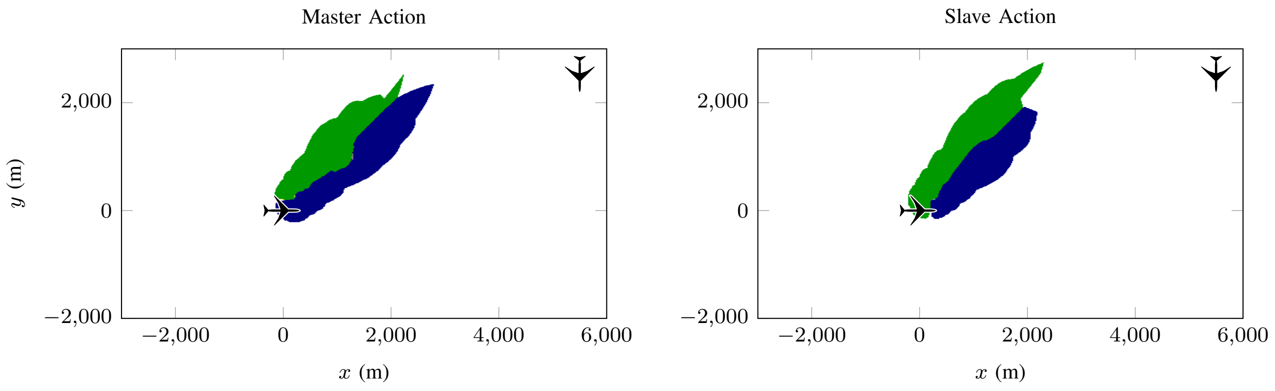
Table II summarizes the optimal policy parameters used in this paper.

TABLE II: Reward parameters for optimal policy

Parameter	Value
R_{NMAC}	−10,000
R_{Close}	−10,000
R_{Maneuver}	−3
R_{Strength}	−0.005



(a) Head-on geometry



(b) Crossing geometry

⬜ NA 🟩 S 🟦 SL ⬛ L 🟨 R 🟩 SR

Fig. 2: Example policy

D. *Policy Generation*

An optimal policy is a mapping from states to joint actions that, when followed, results in maximizing the expected sum of rewards. As in prior research on collision avoidance [11], policies are represented using a "score" table that associates a numeric value with each state and action pair. This numeric value, denoted $Q^*(s, a)$, is the expected discounted sum of rewards when starting in state s, executing action a, and then continuing with the policy represented by Q^*. An optimal action a from a given state s is one that maximizes $Q^*(s, a)$.

Given a transition model and reward function, we can compute Q^*. The Bellman equation states that

$$Q^*(s, a) = R(s, a) + \sum_{s' \in S} T(s' \mid s, a) \max_{a' \in A} Q^*(s', a') \qquad (5)$$

where R and T are the reward and transition functions, respectively. A dynamic programming algorithm known as value iteration can be used to compute Q^*. In this algorithm, $Q^*(s, a)$ is initialized to 0 for all s and a. The equation above is then used to update the estimate of the state-action value for all of the state-action combinations. The update process is iterated until convergence [12], which requires about ten hours on eight Intel Xeon CPU E5-2650 cores running at 2.6 GHz.

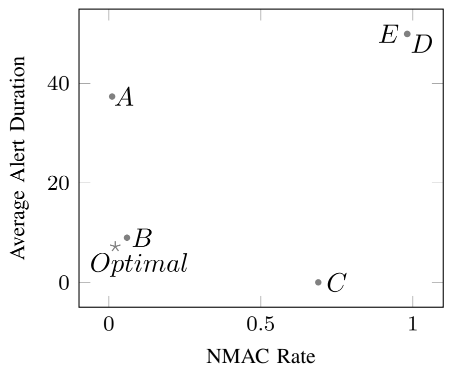
Figure 2 shows a policy for a slice through the state space where both aircraft are moving at 40 m/s. The heading of the slave is indicated by the aircraft figure in the top right corner of the plots. If the slave is located 100 m in front and 100 m to the right of the master with the heading orientations as shown in Figure 2a, then both aircraft would be commanded SR. For the same scenario but with the heading orientations as shown in Figure 2b, both would be commanded SL.

Since the future airspace will likely include aircraft running collision avoidance systems that have been optimized differently, a variety of different policies were generated by varying the reward parameters. Table III shows the reward parameters that were used to generate six different categories of policies. Figure 3a shows the tradeoff between the alert duration during an encounter and the NMAC rate as evaluated on 10,000 encounters. Figure 4 shows the policy for an adversarial policy, Policy E, in a head-on geometry encounter. The behavior is significantly different from the optimal policy because the adversarial policy alerts everywhere and selects actions that bring the aircraft closer together.

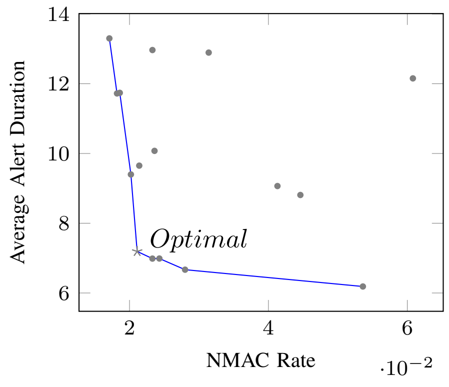
The optimal policy was selected by doing a sweep of reward values and finding the combination that effectively balances safety and efficiency. Each tested policy was run

TABLE III: Reward parameters for example policies

Name	Reward Structure	R_{NMAC}	R_{Close}	R_{Maneuver}	R_{Strength}
Optimal	Best safety vs. efficiency	−10,000	−10,000	−3	−0.005
A	No penalty for maneuvers	−10,000	−10,000	0	0
B	Large penalty for disruptive maneuvers	−10,000	−10,000	−3	−5
C	No penalty for NMAC	0	0	−3	−0.005
D	Reward for NMAC	10,000	10,000	−3	−0.005
E	Reward for NMAC, No penalty for action	10,000	10,000	0	0



(a) Example Policies



(b) Optimal Policy reward sweep

Fig. 3: Various policies NMACs vs alerts

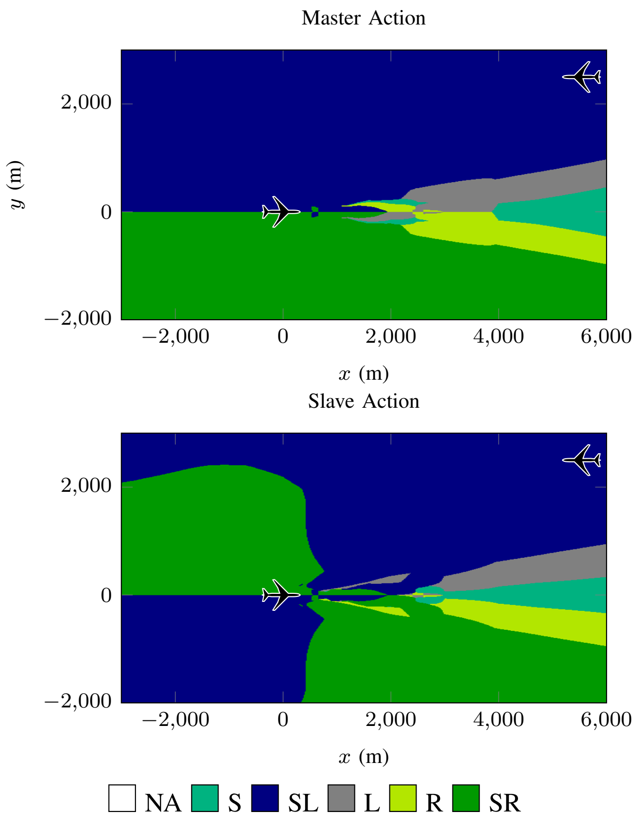


Fig. 4: Example head-on geometry, adversarial policy, E

through the same 10,000 encounters and Figure 3b shows their aggregate NMAC and alert metrics. The selected optimal policy is denoted by "⋆" in Figure 2.

III. Action Selection with Coordination

At each timestep, each aircraft has to select the best action to execute and select an outgoing coordination message (OCM). A coordination message is how the two aircraft communicate to each other and enforces coordination. The selection process requires four inputs as shown in Figure 5 where the selection process is outlined as follows:

1) Receive incoming coordination message (ICM) and ignore inconsistent actions
2) Query coordination table and ignore inconsistent joint actions
3) Select best aircraft action and outgoing coordination message (OCM) based on intruder model

A full description of the various inputs and steps are presented in the following subsections.

A. *ICM and Ownship Actions*

The ICM is from the previous time step and instructs the aircraft to "do not turn left (DTL)," "do not turn right (DTR)," or "either" based on the previous steps coordination outcome. This is the first input considered when selecting the current timestep action and OCM. Using this input, the current state score table is adjusted to ignore inconsistent actions. If we are the master and have received a DTL ICM, the resultant current state score table is the gray area shown in Figure 6a.

B. *Coordination Table*

A coordination table is formatted like a policy except instead of being a mapping from state to action, the coordination table is a mapping from state to whether the two aircraft maneuver in the same or different directions.

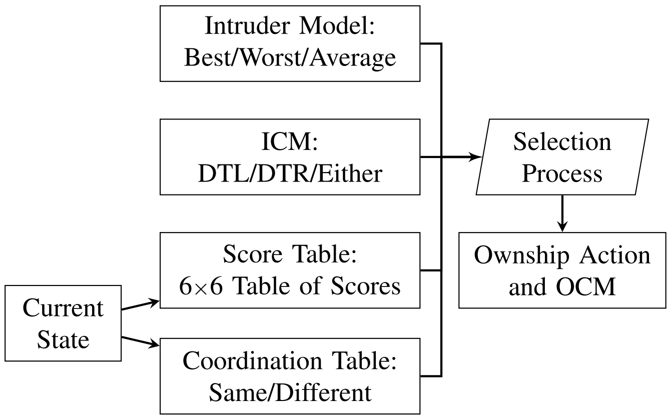

Intruder Model: Best/Worst/Average
ICM: DTL/DTR/Either
Score Table: 6×6 Table of Scores
Current State
Coordination Table: Same/Different
Selection Process
Ownship Action and OCM

Fig. 5: Action-selection process with a coordination table

		Master					
		SL	L	S	NA	R	SR
Slave	SL			▨	▨	▨	▨
	L			▨	▨	▨	▨
	S			▨	▨	▨	▨
	NA			▨	▨	▨	▨
	R			▨	▨	▨	▨
	SR			▨	▨	▨	▨

(a) Limited by a DTL coordination message

		Master					
		SL	L	S	NA	R	SR
Slave	SL			▨	▨		
	L			▨	▨		
	S			▨	▨	▨	▨
	NA			▨	▨	▨	▨
	R			▨	▨	▨	▨
	SR			▨	▨	▨	▨

(b) Limited by a DTL coordination message and the coordination table

Fig. 6: Master current state score table

Generation: The best way to construct a coordination table is an area for further research, but this paper uses a coordination table derived from the optimal score table. First, the score of any action that includes a $0°$ bank angle command or NA is set to $-\infty$ because being commanded to go straight or clear of conflict provides no information about the directions of the joint action. Next, the joint action with the highest score is found for each state. The joint actions are converted to -1 and 1 in the coordination table depending on whether the actions that comprise the joint action are in different or in the same direction. Figure 7 is a plot of a coordination table through a slice of the state space for chasing, head-on, and crossing geometries.

Usage: At each timestep, the aircraft use their current state to query the coordination table and adjust the current state score table to ignore inconsistent joint actions. If we continue the example from earlier where the master current state score table is limited by a DTL ICM then the resultant current state score table of a 1 coordination table output is the gray area shown in Figure 6b.

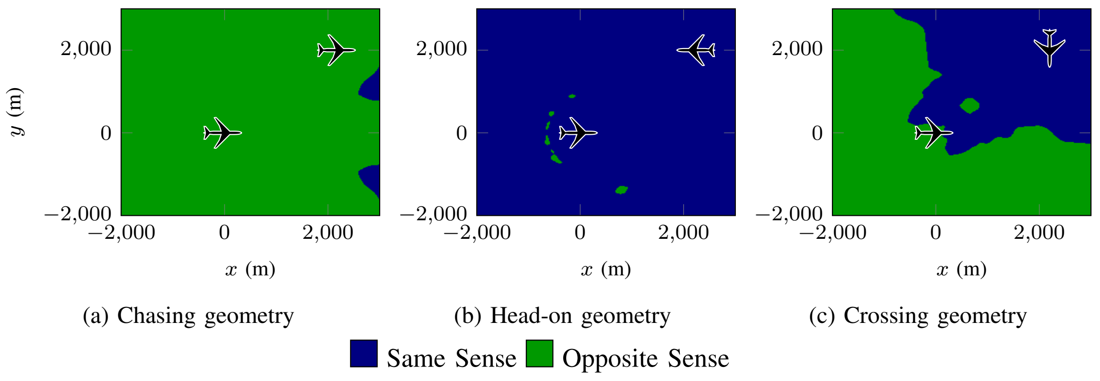
C. Intruder Model's Influence on Action and OCM Selection

It is assumed that the aircraft do not have knowledge about the other aircraft's policy but each aircraft does obey all of the coordination messages and coordination table outputs. Three intruder models were evaluated: best case, worst case, and average case. The process to select an action and coordination message for each intruder model are as follows:

Best Case. It is assumed that the intruder will take their action from the optimal joint action and therefore the current aircraft will take their action of the optimal joint action. Using the selected action and the coordination table output, the current aircraft selects the OCM. Using the previous example, if we are the master aircraft and have the current state score table shown in Figure 8a where the darker the entry, the larger the score (the white entries are ignored based on inconsistencies with the coordination messages and table) then the optimal joint action is for the master to follow NA and the slave to turn SR. For this example, the OCM for the master and slave are set to DTL. It is important to note that at each time step, only the master's coordination messages will be kept.

Worst Case. It is assumed that for any action the current aircraft takes, the intruder will take the corresponding action from the joint action with the lowest score. Therefore, the current aircraft will take their action from the joint action that maximizes the minimum joint action score and send the OCM that captures this action. For example, if the current ownship is the master, the coordination message was "either," and the coordination table returned 1 then Figure 8a is a representation of the available current state score table where the darker the entry, the larger the score (the white entries are ignored based on inconsistencies with the coordination table). Now, we can split this into two tables as seen in Figure 8b and find the minimum score for each master action and OCM. The entries in Figure 8b with "$\star$" show the worst score action the slave can take for each master action and the final row shows the minimum values for each master action. To maximize the minimum score, the master will choose a SR action as denoted by "$\star\star$" and the OCM for the master and slave are set to DTL.

Average Case. It is assumed that for any action the current aircraft takes, the intruder can take any action. Therefore, the current aircraft will take the action that maximizes the average joint action score and send the OCM that captures this action. For example, if the current ownship is the master, if the coordination message was "either," and the coordination table returned 1 then Figure 8a is a representation of the current state score table where the darker the entry, the larger the score (the white entries are ignored based on inconsistencies with the coordination table). Now, we can split this into two



(a) Chasing geometry (b) Head-on geometry (c) Crossing geometry

Fig. 7: Example coordination tables

		Master					
		SL	L	S	NA	R	SR
Slave	SL						
	L						
	S						
	NA						
	R						
	SR						

(a) Limited current state score table

		Master								
		SL	L	S	NA		S	NA	R	SR
Slave	SL		⋆			S			⋆	
	L			⋆		N				
	S	⋆				R		⋆		
	NA				⋆	SR	⋆			⋆
	Min					Min				⋆⋆

(b) Worst case intruder model

		Master								
		SL	L	S	NA		S	NA	R	SR
Slave	SL					S				
	L					N				
	S					R				
	NA					SR				
	AVG	⋆⋆				AVG				

(c) Average case intruder model

Fig. 8: Current state score table limited by coordination table

tables as seen in Figure 8c and find the average score for each master action and OCM. The largest average score is selected as the master action and the corresponding OCM is sent. For the example in Figure 8c, the master would select SL as denoted by “⋆⋆” and the OCM for the master and slave are set to DTR.

D. *Base Case*

To understand the performance of a coordination table, a base case was formulated that does not use a coordination table. Any time the base case is used, it is the approach of both aircraft. At each decision point, each aircraft looks up their current state score table and naively follows their action of the joint action with the maximum score value.

IV. Simulation Framework

A set of experiments were run to evaluate the effectiveness of a coordination table in improving safety and efficiency. The experiments use encounter scenarios where the two aircraft start at a random position on a circle with radius between 1 km and 2 km. The aircraft are oriented such that they point directly at the center of the circle. The master is initialized to have a Gaussian distributed velocity with mean 40 m/s and standard deviation 1 m/s. The velocity of the slave was initialized such that, with 0° bank angle every encounter will generate an NMAC.

The dynamics used in simulation are different from what was used for solving the policy to reduce the risk of overfitting and prove the robustness of including a coordination table. The dynamics equations during an encounter are solved using an ODE solver that incorporates Gaussian noise in acceleration and bank angle. The aircraft receives a commanded bank angle that is followed using a simulated PID controller that is tuned to achieve the commanded bank angle through

$$\ddot{\phi} = -2\omega_n\dot{\phi} + w_n^2(\phi^{\text{com}} - \phi) \qquad (6)$$

where ω_n is a constant related to available control power and ϕ^{com} is the commanded bank angle. The experiments use ω_n set to 0.2. To account for measurement errors during simulation, zero-mean Gaussian noise is added to the commanded bank angle. For NA advisories, the standard deviation is set to 5°. All other advisories have a standard deviation of 3°.

In each experiment, the aggregate metrics are computed from the same 10,000 encounters. All experiments assume the pilots always respond immediately to the given advisory. A decision is made every five seconds in each of the encounters and the OCM is the ICM for the next timestep.

V. Results

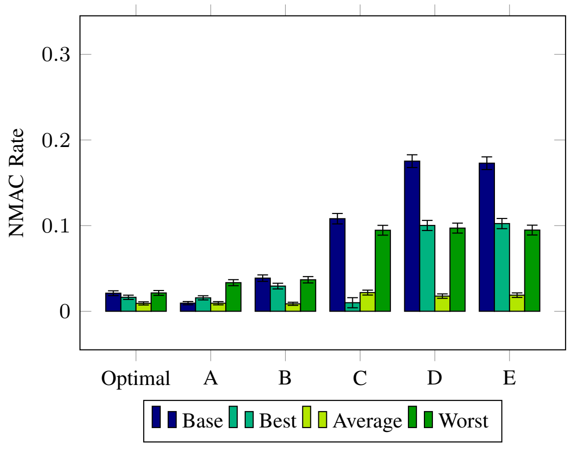
Different combinations of master and slave policies as well as intruder modeling were run on the 10,000 test encounters. Figure 9a shows the resulting NMAC rate and the 95% confidence interval when the master follows the optimal policy and the intruder follows the different policies indicated on the horizontal axis. For each pairing, the master uses the intruder model indicated by the key and the slave always uses the average case intruder model.

The results show that the only time safety degrades with coordination is when the optimal master is interacting with Policy A. This is expected because the policy used to generate the coordination table is not as safe as Policy A, which prioritized an exceptionally low NMAC rate at the expense of a high alert rate. It is expected that the optimal master against Policy B would have minimal improvement in safety because both policies were relatively safe without a coordination table. An optimal master against Policy C, Policy D, and Policy E are originally more unsafe and using the coordination table with the master and slave using average case intruder modeling made these encounters 80%, 90%, and 89% more safe, respectively.

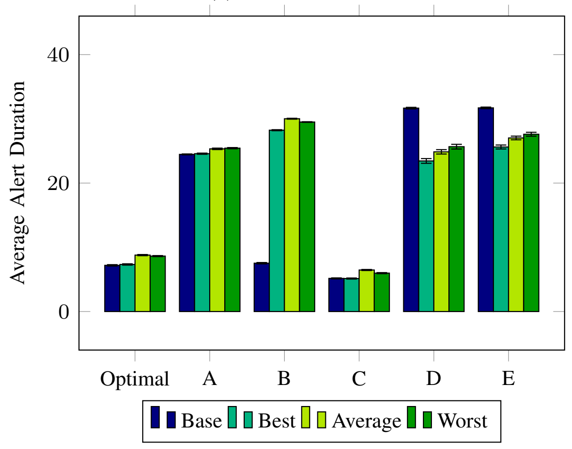
Figure 9b shows the resulting rate of alerts and 95% confidence interval. Overall, the alert rates remain relatively constant with the incorporation of the coordination table except when the optimal master is interacting with Policy B. This result is expected because Policy B was trained to be overly efficient at the expense of safety, which the coordination table helps rectify.

Further, these results are potentially too good which is related to the instantaneous pilot response. Therefore, a delayed pilot response model was employed in which pilots are assumed to be non-responsive for five seconds following an initial advisory, after which they become responsive for the remainder of the encounter. This leads to the rate of NMAC results and 95% confidence intervals presented in Figure 9c where the master follows the optimal policy and the slave follows the different policies indicated on the horizontal axis. For each pairing presented, the master uses the intruder model indicated by the key and the slave always uses the average case intruder model. The results show that adding the coordination table with the master and slave using average case intruder modeling make encounters with Policy D and Policy E 57% and 58% more safe, respectively.

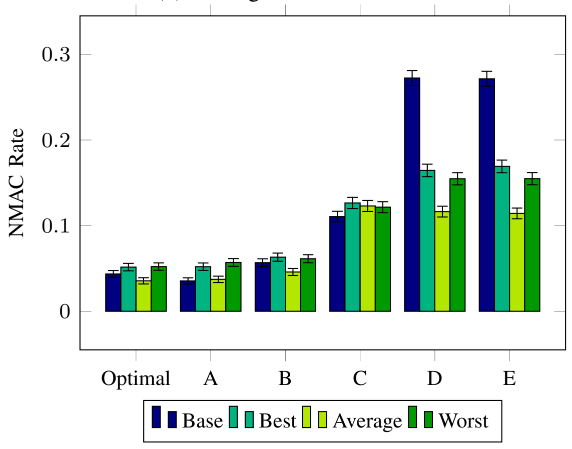
It is believed that the overall positive performance of the average case intruder modeling is due to a lack of overfitting compared to best and worst case which are overly biased to extreme actions. It is also possible that the positive performance occurs because the intruder models for the master and the slave match. The same simulations were run where the intruder models for the master and slave match for best and worst case as shown in Figure 10. The results show that having the modeling match was not the reason that average case performed so well since average case still outperforms the best and worst case.



(a) NMAC rate



(b) Average alert duration



(c) NMAC rate with delayed pilot response

Fig. 9: Results for optimal master with intruder model indicated by the key and various slaves with average case intruder model

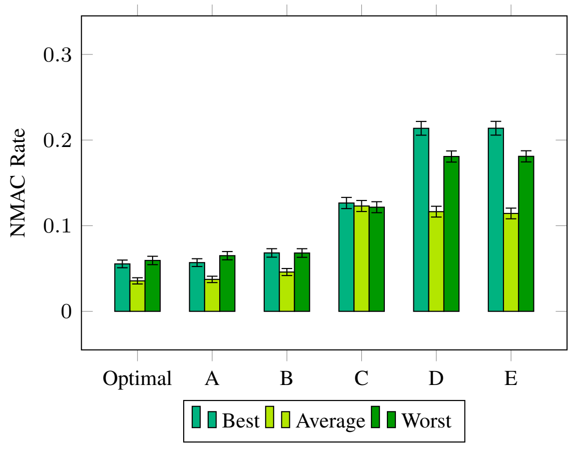


Fig. 10: NMAC rate for optimal master and various slaves with intruder model indicated by the key and delayed pilot response

These more realistic results still have the average case intruder model performing the best and almost always outperforming the cases without a coordination table. The margin is expected to be noise and due to only running 10,000 encounters. For Policy D and Policy E, there is a reward for NMAC and a large increase in safety as anticipated.

Figure 11 shows the same encounter with different policies where the two aircraft start from the • and one aircraft always has the optimal policy against various intruder policies. The colors of the trajectory relate to the actions taken by the aircraft at that time step. The top row shows the interaction without the coordination table which always has an NMAC. The bottom row shows the interaction with the coordination table and the encounter is always resolved. For each encounter, the coordination table has limited the available actions which has resulting in more consistent advisories which lead to safer trajectories. Also, due to intruder modeling, the aircraft often start and end maneuvering earlier in the encounter. The stars on each trajectory show the point of closest approach for each encounter, and illustrate significant improvement through the use of a coordination table. The encounters were run where the master and slave model the intruder as average case and the pilot response is immediate. The plots show that a coordination table adds significant separation. Further, the coordination table is able to reroute adversarial aircraft to prevent collision as seen in Figure 11f.

An additional experiment was run that looks at the effect of a coordination table when two adversarial aircraft are in an encounter. The same 10,000 encounters were run on two Policy E aircraft. There is a 25% increase in safety between the base case and when the master and slave use a coordination table and average case intruder modeling.

VI. Conclusion

This paper uses dynamic programming to create a coordination table that can be used with a bank of policies to create safe encounters. The simulation results suggest that with appropriate intruder modeling, a coordination table can improve safety while maintaining alert rates. The benefit of a coordination table is especially apparent when a safe policy is interacting with an adversarial policy with around a 57% safety increase and between two adversarial policies with around a 25% safety increase.

The initial results of this research are promising and promote several avenues for further research. Currently, only one method has been used to create a coordination table. Future work will explore other methods for creating a coordination table, such as reinforcement learning. The fact that using average intruder modeling with a coordination table performs so well suggests that the coordination table may capture the essential elements of the two-agent problem and that the master collision avoidance system may not need to plan with joint actions. To further explore this hypothesis, we would like to explore more categories of policies such as those used by ACAS Xu, JOCA, or MIDCAS. Finally, the current encounter generation model is designed to produce encounters that should end in NMAC rather than encounter that represent the actual airspace. Future work will address this by testing the effect of a coordination table in realistic encounters.

The software for computing the coordination tables and performing the experiments is found at:
https://github.com/sisl/HorizontalCoordUAVs

Acknowledgment

The authors would thank Kyle Julian for his help with the flight dynamics. The authors wish to thank Neal Suchy for his support. This work is sponsored by the Federal Aviation Administration under Air Force Contract #FA8721-05-C-002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

[1] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

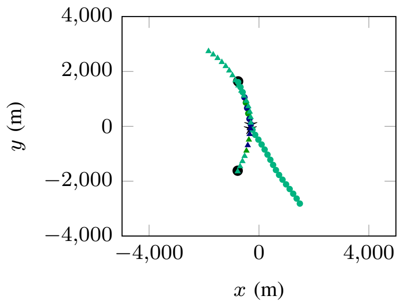
[2] RTCA, *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II)*, DO-185B, Jun. 2008.

[3] 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.

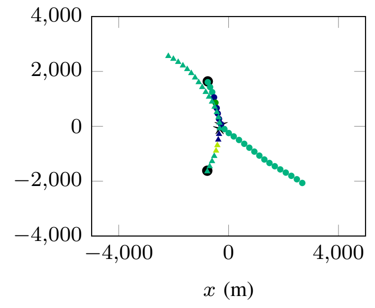
[4] *Introduction to TCAS II version 7.1*, U.S. Department of Transportation Federal Aviation Administration, 2011.

[5] M. P. Owen, "Enhanced uas separation capability using markov decision processes," in *Digital Avionics Systems Conference (DASC)*, 2016.

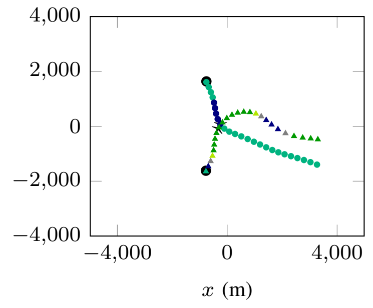
[6] R. H. Chen, A. Gevorkian, A. Fung, W.-Z. Chen, and V. Raska, "Multi-sensor data integration for autonomous sense and avoid," in *AIAA Infotech at Aerospace Technical Conference*, 2011.



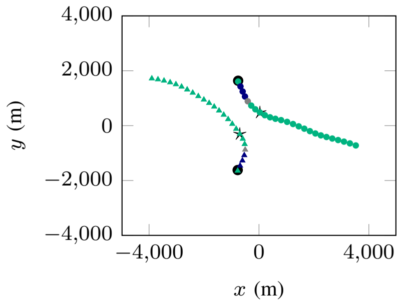
(a) Optimal vs. Optimal without table



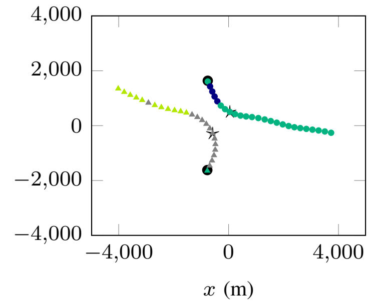
(b) Optimal vs. Policy B without table



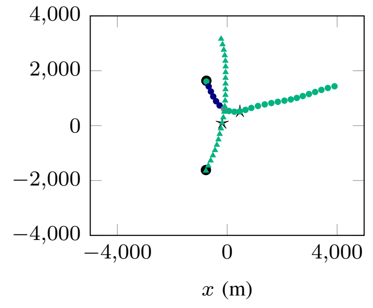
(c) Optimal vs. Policy E without table



(d) Optimal vs. Optimal with table



(e) Optimal vs. Policy B with table



(f) Optimal vs. Policy E with table

🟩 NA/S ■ SL ■ L 🟩 R 🟩 SR

Fig. 11: Example trajectories without and with a coordination table

[7] J. Pellebergs and S. Aeronautics, "The MIDCAS project," in *International Congress of the Aeronautical Sciences*, 2012.

[8] H. Y. Ong and M. J. Kochenderfer, "Short-term conflict resolution for unmanned aircraft traffic management," in *Digital Avionics Systems Conference (DASC)*, 2015.

[9] E. Mueller and M. J. Kochenderfer, "Multi-rotor aircraft collision avoidance using partially observable Markov decision processes," in *AIAA Modeling and Simulation Technologies Conference*, 2016.

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

[11] M. J. Kochenderfer, "Optimized airborne collision avoidance," in *Decision Making under Uncertainty: Theory and Application*, MIT Press, 2015.

[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ: Wiley, 2005.

[13] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2007.

[14] R. Bellman, "On the Theory of Dynamic Programming," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, no. 8, p. 716, 1952.

[15] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

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