UAV Coordination Tables via Learning from Various Logic Tables

Rachael E. Tompa

Stanford University, Stanford, CA, 94305

The risk of unmanned aerial vehicle (UAV) collisions continues to rise as the number of UAVs in the national airspace increases. Many institutions are working on various anti-collision softwares that will implement various logics on UAVs. The logics will produce horizontal maneuver commands/actions to avoid collision during encounters. Due to the nature of horizontal maneuvers, there needs to be a way to ensure that all of the UAVs maneuver safely with one another regardless of logic. One method to do this is to query a coordination table before performing any maneuvers. A coordination table can be created through combining various logics optimizing for safety and efficiency, and returning if the UAVs should maneuver in the same direction, different directions, or if one is going in a neutral direction for specific UAV spacing and headings.

I. Introduction and Related Work

Currently, the number of unmanned aerial vehicles, UAVs, in the national airspace is rapidly increasing which has increased the risk of UAV collisions. Further, the anticipated region for UAVs to fly is a limited altitude band which means that anti-collision efforts must occur in the horizontal plane rather than the vertical. One way to mitigate UAV collisions is through equipping UAVs with anti-collision software. Previous work at Stanford has started this process by modeling the problem as a Markov decision process (MDP) and solving for optimal policies. This representation uses value iteration to solve for the optimal policy and considers multiple UAV encounters by decomposing the encounter, solving each dual UAV encounter, and then recombining all encounters to create the overall optimal policy [1]. The policies provide each UAV a bank angle or clear of conflict command to advise the pilot on how to proceed to optimally avoid collisions. The policies consider different pilot responses, initial conditions, and the number of UAV per encounter. Simulations of this method have provided positive results and collision mitigation, but other institutions such as NASA and the FAA are also developing their own policies via different methods, therefore resulting in potentially different logics.

It is expected that eventually all UAV's will include collision detection and mitigation but it is not expected that only one method of mitigation will be used. Since it is proposed that a variety of logic will be implemented on various UAVs it is important to have a method to coordinate all of the different logics to ensure that even with different logics, the UAVs are avoiding collision. This is not needed with large aircraft because during large aircraft encounters the same logic is always used and during large aircraft encounters the maneuver commands are in the vertical plane and therefor the maneuver directions should always be opposite or neutral (one aircraft goes up and one aircraft goes down or only one aircraft maneuvers up/down). When maneuvering in a horizontal plane, the maneuver directions are not always opposite. An example of this problem is better outlined in Figure 1. One way to address the problem for UAVs is to create a standard coordination table that can be queried before any maneuvers are performed.

A coordination table is a mapping of UAV headings and positions to the safe maneuver pairings (same heading, opposite heading, neutral heading). This project is the first attempt at making a coordination table for UAV encounters and serves as a proof of concept. This project creates coordination tables for dual UAV encounters by running various policies through multiple encounters, quantifying the optimal policy per encounter, combining the optimal policies into a super policy using nearest neighbors, and uses the super policy to solve for coordination commands.

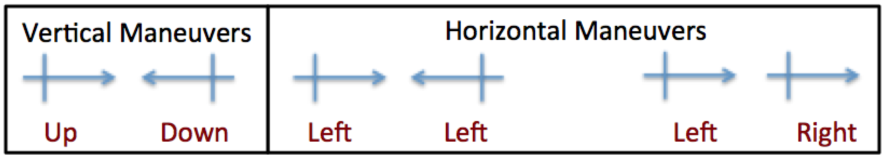


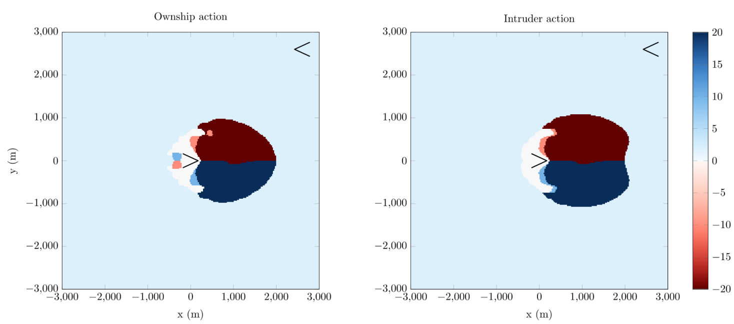
Figure 1: Vertical vs. horizontal maneuvers.

II. Dataset

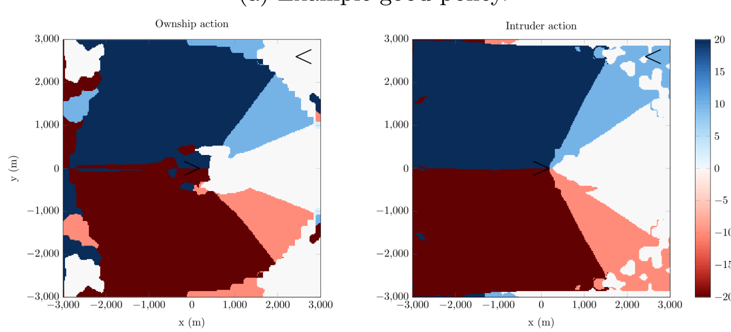
Policies, metrics, and encounters were needed to create a coordination table. The MDP in [1] defines states(s), actions(a), a reward function(R), and a transition function(T). The MDP is solved with value iteration and the optimal action is solved for using the Bellman equation,

$$U^*(s) = \max_{a \in A} \left[R(s,a) + \sum_{s' \in S} T(s' \mid s,a) U^*(s') \right]. \tag{1}$$

The reward function penalizes conflict, minimum separation, closeness, and actions. To create an assortment of logics, the weights of the different penalties were adjusted and the corresponding policy was found. Example policies are shown in Figure 2. These plots represent what the labeled UAV (located at the centroid with the heading as displayed by the center arrow) would be commanded to do if the other UAV was located at any position on the plot (with the heading as display by the top right arrow). The metrics used



(a) Example good policy.



(b) Example bad policy.

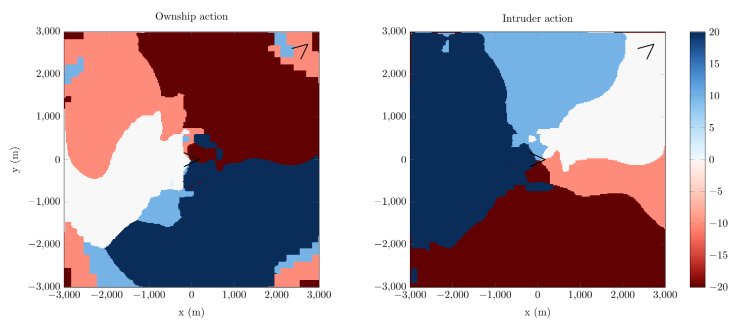
Figure 2: Example policies used in coordination table generation.

to determine the quality of a policy were safety and efficiency. A safer policy and therefore better policy will result in fewer collisions and a more efficient policy will result in fewer aircraft commands during encounters. It was important to have a variety of policies because it is unknown how good actual implemented logics will be. To test the policies, they are ran on simulated encounters. For the experiments in this project, 100 encounters between two UAVs were created using the generator in [1]. All of the encounters are set up so the initial conditions put the UAVs on a collision trajectory.

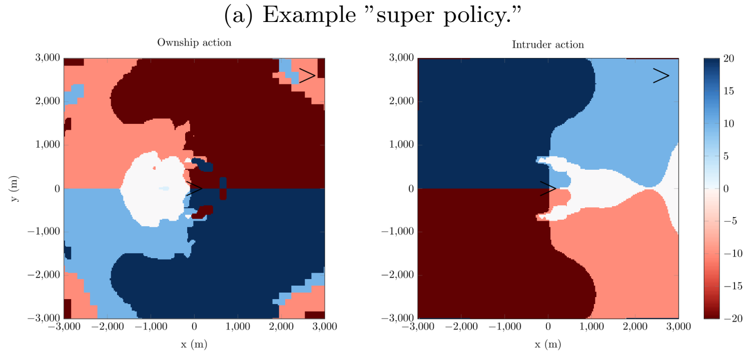
III. Methods

Creating a coordination table was a multi-step process. To begin, the policy generator in [1] was used to create 34 various of policies. Using visual inspection and trial encounter simulations, six unique policies were selected for experimentation. Each of the selected policies were run through the same 100 encounters. During each encounter, the safety and efficiency of the policy was recorded. The overall performance for all 100 encounters for the policy was also recorded. For each encounter, the best policy was selected. The best policy was defined as the policy that had the best safety, where ties were broken by selecting the most efficient policy, where ties were broken by selecting the policy that had the best metrics for all 100 encounters (safety always outweighs efficiency).

Each policy is actually a set of commanded actions for specific nodes of a grid. The grid is seven dimensional and for this project, only x position, y position, and heading were altered. The other components are velocity and pilot response which were held fixed. A "super policy" was made on the same grid system. The "super policy" was made using the encounter to optimal policy mapping. For each grid node of the "super policy", nearest neighbors was used to find the encounter with the closest initial position. The optimal policy for the nearest neighbor encounter was the selected policy for the "super policy" grid node. In this sense, the "super policy" is a combination of all of the policies. Two example "super policies" are shown in Figure 3. The "super policies" are presented in the same fashion as the policies in Figure 2. Next, to



(a) Example "super policy."

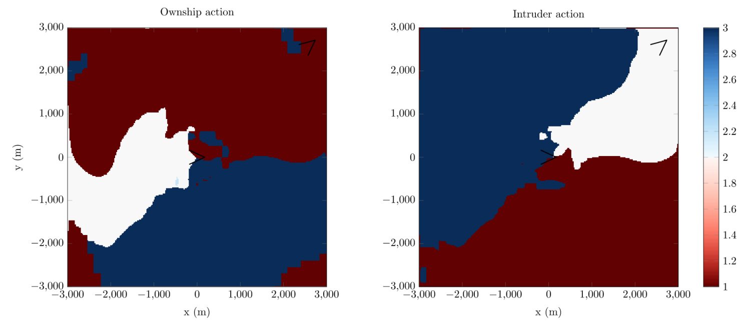


(b) Example "super policy."

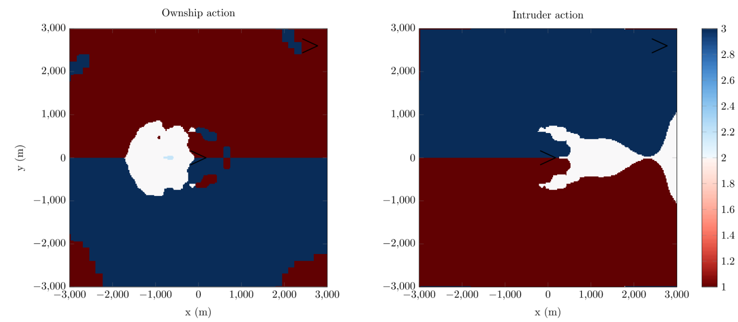
Figure 3: Example "super policies."

convert the "super policy" into a coordination table, all of the actions were adjusted from bank angles to pure directions: left, right, straight/no command. The "super polices" from Figure 3 are shown in pure direction form in Figure 4. In these plots, red is left, blue is right, and white is straight/no command (this matches the trends of Figure 2 and Figure 3).

Finally, combining the ownship and the intruder policies, a coordination table was made. The coordination table shows when the ownship and intruder should perform a maneuver in the same direction, opposite directions, or if at least one UAV is commanded to go straight or no command.



(a) Example super policy.



(b) Example super policy.

Figure 4: Example pure direction plot where red is left, blue is right, and white is straight/no command.

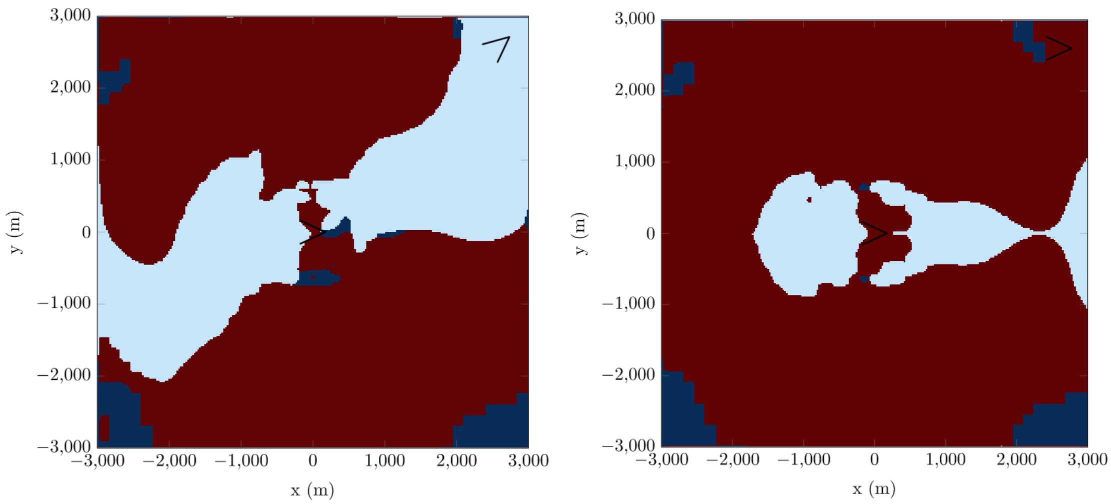
IV. Results and Discussion

The coordination tables for the policies presented in Figure 3 and Figure 4 are presented in Figure 5. In these plots, red is opposite headings, blue is the same heading, and light blue is at least one aircraft has a neutral heading command (straight/no command). This first cut at a coordination table is promising. To begin, for the initial investigation, the regions that have the same headings and regions that have different headings are mostly as expected. When the UAV's are going generally in the same direction, the maneuvers are usually in opposite directions. When the UAV's are going generally in the opposite directions, the maneuvers are mostly in the same direction. Further, since both UAVs in the experiments were set to have the same velocity, the region where at least one UAV has a neutral maneuver command is logical. The neutral region is of particular interest for future iterations. When creating the initial coordination tables the main concern was on when the maneuvers for each UAV should or should not match and therefore, the neutral maneuvers were not really considered. In exploring the coordination tables, it has also become apparent that there needs to be a way to communicate how to respond to neutral actions. Also, as anticipated, many of the coordination tables are symmetric or trend towards a line of symmetry.

Looking at the policies, there are some other inconsistencies. Two are shown in Figure 5a. First, not all of the policies are symmetric. Second, some of the maneuvers are opposite of what is anticipated. It is expected that both the unsymmetrical nature and deviation from anticipated results are due to the state space discretization and the limit of encounters used in the nearest neighbor selection of policies.

V. Conclusion and Future Work

The work for this project is a proof of concept for a larger research project that I am working on. This project successfully showed that a coordination table can be created and has exposed areas that need improvement. To begin, all of the created policies could be run for the encounters to make sure the optimal policy was always selected. If deemed necessary, additional policies could also be created. Further, all of the policies should be run on more encounters, that are more diverse. Currently, the method that [1] uses to create an encounter initializes the UAVs so that a collision is possible at the center and ensures that the



(a) Example super policy.

(b) Example super policy.

Figure 5: Example pure direction plot where red is left, blue is right, and white is straight/no command.

encounter does not begin in a collision. The encounter generator tends to create a circular region of starting positions. The starting positions for the current encounters used are shown in Figure 6. Future encounter

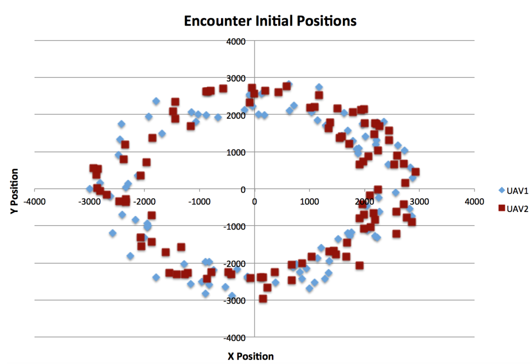


Figure 6: 100 encounter starting positions used for experiments.

generation will add diversity by changing the location of collision, enabling additional initial starting positions that will benefit the nearest neighbor calculation hopefully enhancing the "super policy."

If the additional policies and encounters do not alleviate the unsymmetrical trends and explain the unexpected results, further investigation will be conducted. The initial investigation would look at the effects of the state space discretization. Currently, to contain the number of overall states, the state space is discretized which potentially losses some of the information about the intermediate states not depicted on a grid node. The discretization can be adjusted to better understand the inconsistencies. Once these concerns are addressed the coordination table effectiveness should be measured.

To understand how the coordination table preforms, simulations will be run where the UAVs follow different policies and query the coordination table before maneuvering. This process has two steps. First, currently there is no framework to encounter simulations with different policies. Second, there needs to be a method to query the coordination table. When the coordination table is queried there needs to be a process established for when maneuver pairs do not match what the coordination table dictates. Once it is established that coordination tables can be used to increase safety, they will be expanded to address encounters with more than two UAVs.

References

[1] H. Y. Ong and M. J. Kochenderfer. Short-term conflict resolution for unmanned aircraft traffic management. Proceedings of the 34th Digital Avionics Systems Conference (DASC), Sep. 2015.