

Implicitly Coordinated Detect and Avoid Capability for Safe Autonomous Operation of Small UAS

Swee Balachandran*

National Institute of Aerospace, Hampton, VA, USA

César Muñoz[†] and María Consiglio[‡]

NASA Langley Research Center, Hampton, VA, USA

As the airspace becomes increasingly shared by autonomous small Unmanned Aerial Systems (UAS), there would be a pressing need for coordination strategies so that aircraft can safely and independently maneuver around obstacles, geofences, and traffic aircraft. Explicitly coordinating resolution strategies for small UAS would require additional components such as a reliable vehicle-to-vehicle communication infrastructure and standardized protocols for information exchange that could significantly increase the cost of deploying small UAS in a shared airspace. This paper explores a novel approach that enables multiple aircraft to implicitly coordinate their resolution maneuvers. By requiring all aircraft to execute the proposed approach deterministically, it is possible for all of them to implicitly agree on the region of airspace each will be occupying in a given time interval. The proposed approach lends itself to the construction of a suitable feedback mechanism that enables the real-time execution of an implicitly conflict-free path in a closed-loop manner dealing with uncertainties in aircraft speed. If a network infrastructure is available, the proposed approach can also exploit the benefits of explicit information.

Nomenclature

ADS-B	Automatic Dependent Surveillance-Broadcast
BVLoS	Beyond Visual Line of Sight
DAA	Detect and Avoid
DAIDALUS	Detect and Avoid Alerting Logic for Unmanned Systems
RRT	Rapidly Exploring Random Trees
TCAS	Traffic Alerting and Collision Avoidance System
UAS	Unmanned Aerial Systems

I. Introduction

Unmanned Aerial Systems (UAS) are growing to be indispensable tools in today's society. Their small size, agility, and capability to carry a sophisticated suite of onboard sensors make them extremely useful in multiple domains such as surveillance, agriculture, inspection, and package delivery. NASA's Unmanned Aerial System (UAS) Traffic Management (UTM) project aims at enabling near-term, safe operations of small UAS vehicles in uncontrolled airspace, i.e., Class G airspace. In the UTM Concept of Operations,¹ UAS will use "onboard detect and avoid systems to avoid other traffic, adverse weather, terrain, and man-made and natural obstacles."

Years of research, development, and testing efforts have been invested into the Detect and Avoid (DAA) technology for the integration of large UAS in non-segregated airspace. However, DAA algorithms for large UAS were designed to support compliance with existing see-and-avoid regulations for manned aviation, and hence, they support a concept of operations in which a pilot-in-command is responsible for the execution of DAA maneuvers as well as return to path maneuvers. In other words, DAA algorithms for large UAS are designed to detect potential conflicts with other aircraft and provide maneuver guidance and traffic alerts to the pilot-in-command.²

The operational environment and the specific challenges faced by small UAS operating at low altitudes in uncontrolled airspace will require new algorithms specifically designed for them. In particular, it is essential that the DAA

*swee.balachandran@nianet.org

[†]cesar.a.munoz@nasa.gov

[‡]maria.c.consiglio@nasa.gov

algorithms for small UAS also take into account obstacles and geofence constraints in addition to other traffic. However, if all aircraft operating in an airspace start using various resolution maneuvers without a coordination strategy, the resolutions they execute can be ineffective and lead to secondary conflicts. In a multiple encounter scenario, satisfying separation constraints may require explicit communication of the resolution strategy of each aircraft. Explicitly coordinating maneuvers among conflict aircraft require components such as a reliable communication infrastructure and standardized software protocol for information exchange. These additional components can significantly increase the cost of small UAS and UAS operations. Furthermore, for low altitude UAS operations, in addition to coordinating maneuvers when avoiding other traffic, each vehicle must satisfy any available keep-in and keep-out geofence constraints.

Several approaches have been proposed for ground-based and, more recently, airborne separation assurance systems for manned aircraft.³ Airborne technology for collision avoidance such as TCAS (Traffic Alerting and Collision Avoidance System)⁴ has been successfully adopted by the commercial aviation industry. TCAS II, the second generation of TCAS, issues aural and visual alerts that direct pilots to maintain or increase vertical separation with intruders that are considered collision threats. The TCAS II system is based on an interrogation mechanism between transponders onboard the aircraft and a set of distance and time thresholds that determine the type of alerting and resolution advisory. A new generation of collision avoidance system, called ACAS X (Advance Collision Avoidance System), has been recently proposed.⁵ In contrast to TCAS II, the alerting and advisory logic on ACAS X is based on numerical look up tables optimized with respect to a probabilistic model of vehicles operating in the airspace. For large, remotely piloted UAS, NASA has developed a detect and avoid system called DAIDALUS (Detect and Avoid Alerting Logic for Unmanned Systems).⁶ DAIDALUS logic uses a well-clear volume that is defined on distance and time functions similar to those used in the TCAS II resolution advisory logic.² Multiple aircraft trying to resolve conflicts using one of the above collision avoidance solutions will have to explicitly coordinate the resolution strategies to successfully avoid loss of separation. For example, manned aircraft equipped with TCAS use a Mode S transponder to coordinate resolution maneuvers. The coordination mechanism that are currently used by TCAS and ACAS X poses several challenges in multiple threat encounters.⁷

Several approaches have been proposed to handle the interaction between aircraft when solving a conflict. Some of these approaches rely on explicit exchange of intent information^{8–11} or on an explicit and temporary delegation of responsibility for separation.¹² Richards et al.¹³ explore a centralized approach based on mixed-integer linear programming. The generated trajectories have to be communicated to each UAS using a communication network. Execution of the path will require additional feedback mechanism to cope with uncertainties. Ong et al.¹⁴ explore both centralized and decentralized solutions that utilize Multi-agent Markov Decision Processes (MMDP) to coordinate resolutions between UAS. Unlike the previous approach, the MDP yields an optimal policy and hence thus can address uncertainties during execution. However, the complexity of solving a MMDP increases with the number of agents. Furthermore, the addition of obstacles/geofences requires reconsideration of the MMDP formulations.

The centralized and decentralized approaches both rely on a communication network to broadcast information such as intent, trajectories, and priorities. The primary goal of this work is to formulate a coordinated resolution while avoiding the need to explicitly communicate intent information to other aircraft. The works that are the closest in essence to the primary goal of this paper are those based on state-based implicit coordination. Bilimoria¹⁵ and Narkawicz and Muñoz¹⁶ propose pair-wise implicitly coordinated methods that, in some cases, can be extended to more than two aircraft. For multiple aircraft encounters, Muñoz and Doweck¹⁷ present an algorithm that resolve multiple aircraft conflicts using the vertical dimension. Zeghal,¹⁸ Eby,¹⁹ and Hoekstra²⁰ propose methods that use potential fields to coordinate resolution maneuvers without the need to explicitly communicate information. A drawback of this approach is that potential field methods are prone to local minima especially in encounters involving multiple aircraft. The problem is further compounded when one wants to consider additional constraints such as keep-in and keep-out geofences.

This paper contributes a novel technique to enable multiple aircraft involved in a conflict to coordinate their resolutions with minimal exchange of information, namely aircraft position and velocity. By enforcing each aircraft to execute the proposed algorithm deterministically, all aircraft can agree on the resolution trajectories executed by the other aircraft and thus avoid loss of separation. Additionally, the proposed approach lends itself to the construction of a suitable feedback mechanism that enables the real-time execution of the path in a closed-loop manner dealing with uncertainties in aircraft speed. The safety critical nature of several UAS applications mandates a rigorous understanding of the properties of any algorithm running onboard a UAS. This work shows that by strictly enforcing keep-in geofence constraints at strategic locations along the computed paths, it is possible to guarantee that no two aircraft will encounter a loss of separation with each other. The implementation of the proposed approach makes use of a software library called PolyCARP²¹ for enforcing geofence constraints, whose key safety properties have been

II. Implicit Coordination

For aircraft to coordinate resolutions successfully, each aircraft should (a) come to a consensus on an order that establishes a sense of priority and (b) be aware of the intent of the other aircraft. Information about (a) and (b) can be shared explicitly in the presence of a communication network. In the absence of a communication infrastructure, if a scheme to allow each aircraft to independently arrive at identical conclusions for (a) and (b) can be devised, then all aircraft can coordinate their resolutions successfully without explicitly exchanging any information. This key idea leads to the procedure defined in Algorithm 1.

```

begin
  Inputs: Position and orientation of  $N$  aircraft, function to compute sequencing order, function to compute
    goal location, abstraction of airspace, function to compute flight plan given initial and final positions.
  Compute sequencing order  $S$  for the  $N$  aircraft.
  Compute goal locations for  $N$  aircraft.
  for  $i = 1 : N$  do
    Compute high-level path for the  $i^{\text{th}}$  aircraft in sequence  $S$  from initial to goal location while avoiding all
    previous  $i - 1$  aircraft in sequence  $S$ .
  end
   $n =$  Order corresponding to ownship in sequence  $S$ .
  Compute flight plan for ownship within the  $n^{\text{th}}$  high-level path.
  Execute flight plan computed.
end

```

Algorithm 1: Pseudo code for implicit coordination

There are four main steps that must be performed by each aircraft involved in the conflict. First, a sequencing order is computed for every aircraft. The second step involves computing the goal location for every aircraft. Third, with the sequencing order of all aircraft, their initial positions and their goal locations known, flight plans are computed for all the aircraft to get from their respective initial positions to their goal positions while avoiding any geofences and all previous aircraft in the sequence. Finally, each aircraft executes the plan corresponding to its order in the sequence. If the above steps are performed deterministically, then n instances of the proposed algorithm running onboard n different aircraft independently produces the same set of plans for each aircraft. The above sequence of steps can be accomplished by several ways. One possible instantiation for each of the steps described above is elaborated in the proceeding subsections. Without loss of generality, the illustration presented in this work focuses only on resolutions in the horizontal plan (North-East plan). However, the proposed method can be extended to consider resolutions in three dimensions (North-East-Down). This work assumes the following: (a) all aircraft operating in the airspace use the proposed approach for conflict resolution, (b) each aircraft can observe the position and orientation of the other aircraft in the airspace and (c) the total number of aircraft involved in a conflict remain constant until the conflict is resolved.

II.A. Sequencing order

A sequencing order is a total order over the set of all aircraft involved in the conflict. The sequencing order dictates the order in which the paths are computed for each aircraft. The path computation for the i -th aircraft in the sequence can exploit the knowledge of the $i - 1$ paths computed so far to stay separated from all previous aircraft in the sequence. The computation of the sequencing order may also consider prioritizing aircraft based on maneuverability, similar to the right of way rules used in manned aviation. The construction of a suitable function that assigns a sequencing order is beyond the scope of this paper. For the illustrations considered in the following sections, this paper assumes the existence of a deterministic function that takes as inputs aircraft position, heading and additional information such as aircraft identification to produce a sequencing order.

II.B. Goal generation

The goal location is a position and orientation in 3D space at which all aircraft are conflict free. The computation of the goal location should take into account factors such as aircraft maneuverability, encounter geometries and other

spatial constraints such as geofences and obstacles. This work uses a simple goal location obtained by projecting the current position of the aircraft in the direction of its current heading by a predetermined distance.

II.C. Path generation

Given a sequencing order, an initial and goal position for all the aircraft, the path generation function should provide conflict free resolution trajectories for all aircraft. Furthermore, the path generation must be deterministic so that independent executions of this function by each aircraft yield an identical set of paths. As this approach involves computing multiple paths, the computational complexity increases with the number of aircraft involved in the conflict. Consequently, this work adopts a hierarchical approach to ease the computational burden.

The task of generating a set of paths is decomposed into two parts. First, high-level compact paths are computed over an abstract representation of the airspace using a tree based search method. A high-level path can be viewed as a contiguous sequence of regions in the airspace that must be occupied by an aircraft at various time intervals. To ensure that each aircraft agree on what region it should occupy and that no two aircraft occupy the same region simultaneously without explicit coordination, it is necessary for all aircraft to use a common abstraction of the airspace. This can be achieved by maintaining a database of the abstracted airspace computed a priori. A few examples of suitable abstractions are partitions of the airspace in the form of grid cells or Voronoi regions (see Figure 1). The constructed abstraction must also take into consideration the availability of minimum maneuvering space for vehicles the airspace is intended for. The idea of partitioning the airspace into regions or zones is reminiscent of how air-traffic control partitions the airspace into sectors. Without loss of generality, this work uses Voronoi regions to partition the airspace^a.

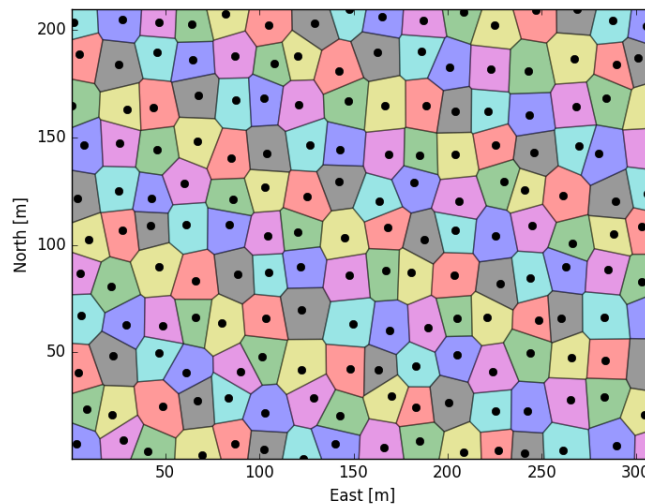


Figure 1: Voronoi tessellation of airspace

A graph data-structure that captures the connectivity of the various regions/cells in the abstracted airspace is constructed. Each region/cell is a node in the graph. Edges connect nodes that represent adjacent regions. Information such as the coordinates of the Voronoi point corresponding to the underlying cell and the vertices of the corresponding Voronoi polygon are stored in a node. Given an initial position and final position in the North-East coordinate frame, the corresponding Voronoi cells are those whose Voronoi points are closest to the given points respectively. A high-level path from the initial cell to the final cell is computed using an A* search algorithm.

The algorithm starts by computing a high-level path for the first aircraft in the sequence. The fundamental assumption behind the high-level paths is that all aircraft transition simultaneously from one cell to the adjacent cell at discrete time steps. The proceeding sections describe how a feedback mechanism allows this assumption to be relaxed during path execution. The high level paths are essentially an over approximation of the space occupied by each aircraft at a given time interval. When computing the high-level path for the i -th aircraft, by keeping track of the nodes occupied by all previous $i - 1$ aircraft at timestep t , it is possible to avoid selecting a neighborhood around these nodes at time timestep t . Consequently, no two aircraft share the same cell at any given time. Finally, each aircraft can use its

^aA rectangular grid like partition can be generated from Voronoi regions by considering Voronoi points that lie on a uniform grid.

corresponding high-level path to compute a feasible low-level flight plan consistent with its dynamics. Note that only the high-level paths need to be computed consistently (deterministically) across all aircraft. Once all aircraft implicitly agree on the sequence of regions they will occupy, each aircraft can compute a feasible path independently. A local planner can be used to connect the Voronoi points of the high-level path to obtain a feasible path from the initial to the final position. This work uses a simple based method based on Rapidly Exploring Random Trees (RRT)²³ as the local planner. Figure 2 illustrates an example of a high-level path and its corresponding low-level flight plan from an initial position to a final position.

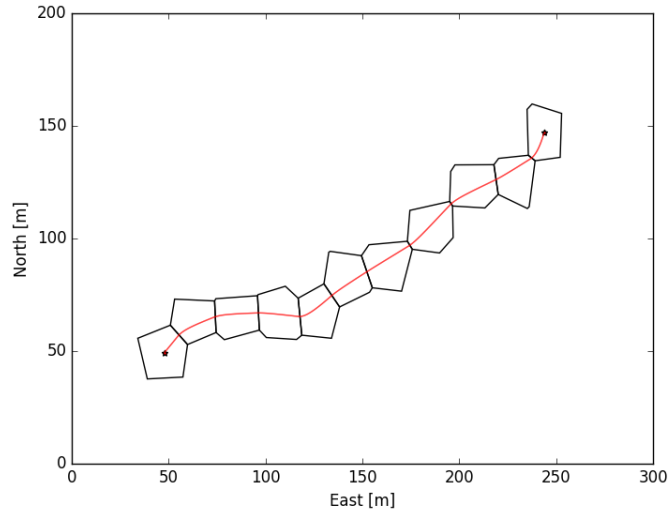


Figure 2: High-level path and corresponding flight plan

II.D. Path execution

The high level planning process assumes that all aircraft transition into the adjacent cell/region simultaneously. This assumption is crucial to ensure that no two aircraft occupy the same cell at any given point in time. However, execution of the computed flight plan in an open-loop setting is not sufficient to cope with uncertainties associated with the speed of the vehicles and hence it is difficult to guarantee that all aircraft will transition into the adjacent cells simultaneously. Fortunately, information about the progress made by the vehicles with respect to their high-level paths can be utilized to construct a feedback policy to govern the path execution.

Consider the following example involving a conflict between three aircraft. Figure 3 illustrates the high level paths obtained for the three aircraft as described in Section II.C. In Figure 3, ‘S’ denotes the start location and ‘G’ denotes the goal location. Assume that the red, green and blue aircraft are the first, second and third in the sequence respectively. Let $\{\rho_1, \rho_2, \dots, \rho_l\}$, $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$ and $\{\beta_1, \beta_2, \dots, \beta_n\}$ denote the sequence of cells in the high-level paths for the red, green and blue aircraft. In an ideal execution, all aircraft transition simultaneously to the adjacent regions (i.e. $(\rho_1, \gamma_1, \beta_1) \rightarrow (\rho_2, \gamma_2, \beta_2) \rightarrow \dots$). When any one aircraft speeds up or slows down, the ideal execution of the high-level path is violated. In those cases, the cells indicated by the asterisks in Figure 3 represent the regions where there is a high likelihood for a collision. Therefore, when each aircraft reaches the cell indicated by ‘?’ on their respective paths, it has to decide if proceeding to the adjacent cell will result in a collision with the other aircraft. If progressing into the adjacent cell is unsafe, the aircraft will be required to enter a holding pattern until it is safe to progress. For rotorcraft, this can be easily achieved by transitioning into a hover whereas for fixed wing vehicles, this can be achieved by transitioning into a loitering mode within the current region. A simple decision tree that enables each aircraft to reason about collision on reaching the ‘?’ cells is provided in Figure 4. This decision making process serves as a feedback mechanism to cope with uncertainties associated with the speed of other vehicles and is applied on a pair-by-pair basis. For example, at the intersection of the red and blue paths in Figure 3, this blue aircraft only has to consider the progress of the red aircraft to decide if its safe to cross into the adjacent region.

The use of polygon-based abstractions of the airspace such as Voronoi regions provide additional benefits. Each high-level path is a sequence of polygons and, therefore, the outer edges of the sequence of polygons can be viewed as a keep-in geofence. Furthermore, the decision tree in Figure 4 can dynamically turn the ‘?’ cells into a keep-in

geofence for any aircraft to avert impending collisions with other vehicles that may cross its path. Consequently, this approach relies on the capability to stay within a prescribed keep-in geofence to ensure no loss of separation occurs.

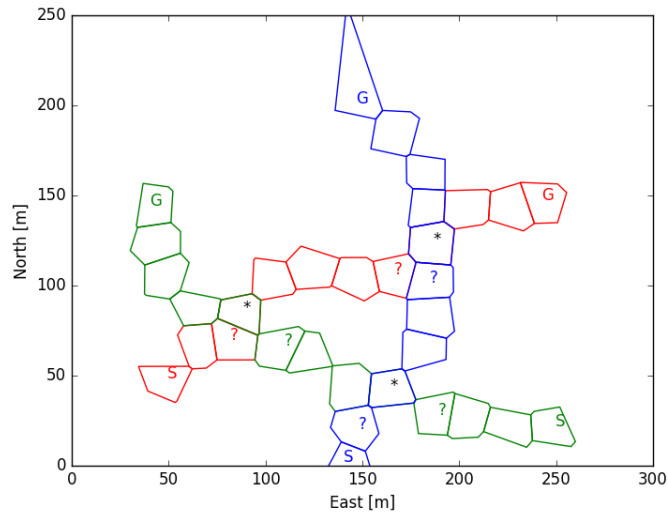


Figure 3: Conflict resolution involving three aircraft

II.E. Results

The following figures illustrate the execution of the implicit coordination approach discussed above. Three aircraft (red, green and blue) are at (50,50), (250,25) and (150,0) respectively with headings 45, 330 and 0 degrees respectively. If the three aircraft continue with the current course, a loss of separation is imminent. Thus, to avoid a loss of separation, each aircraft independently executes the implicit coordination strategy proposed in Algorithm 1. In this work, the following sequencing order is used: red=1, green=2 and blue=3. A Voronoi tessellation is used to construct an abstraction of the airspace (see Figure 1). The implicitly coordinated resolution trajectories of the aircraft are illustrated in Figure 5b. Snapshots of the trajectory execution of all aircraft are illustrated in Figures 6 and 7. Each aircraft is able to go from its initial position to the final position without a loss of separation. Figures 8-9 illustrate the application of the proposed approach on a scenario requiring all aircraft to avoid a keep-out geofence while executing the resolutions.

III. Discussion

An important prerequisite for all aircraft involved in a conflict to successfully apply the proposed approach is to implicitly agree on who all are involved in the conflict without explicit communication. This can be challenging especially if there are uncertainties associated with the initial positions of the individual aircraft. One possible solution is to consider all aircraft operating within a larger virtual geofence. The assumption that all aircraft can observe the position and orientation of the other aircraft in the airspace requires surveillance capability onboard each aircraft, e.g., inexpensive commercial off-the-shelf ADS-B sensors for small UAS.^{24b} Alternatively, the presence of a central server such as the UTM service¹ can also facilitate this assumption. An important challenge of the proposed approach is how one deals with new aircraft entering into a conflict with aircraft that have already started executing the coordinated resolutions. The illustrations presented in this paper resolve conflicts by using 2D trajectories. The proposed approach can also be extended to 3D. However, the complexity of computing paths increases with the dimension of the search space. The goal of this work was to develop a solution to enable coordination without explicit communication. Nevertheless, the proposed approach can easily exploit the benefits of a communication infrastructure, if available. In the presence of a communication infrastructure, this approach can be executed in a centralized fashion where a server performs the computation and distributes the high level paths to the individual aircraft. Alternately, this approach can also be used in a decentralized fashion having the individual aircraft perform the computations.

^bLong-term, sUAS may not be allowed to carry ADS-B out, due to certification issues and frequency congestion.

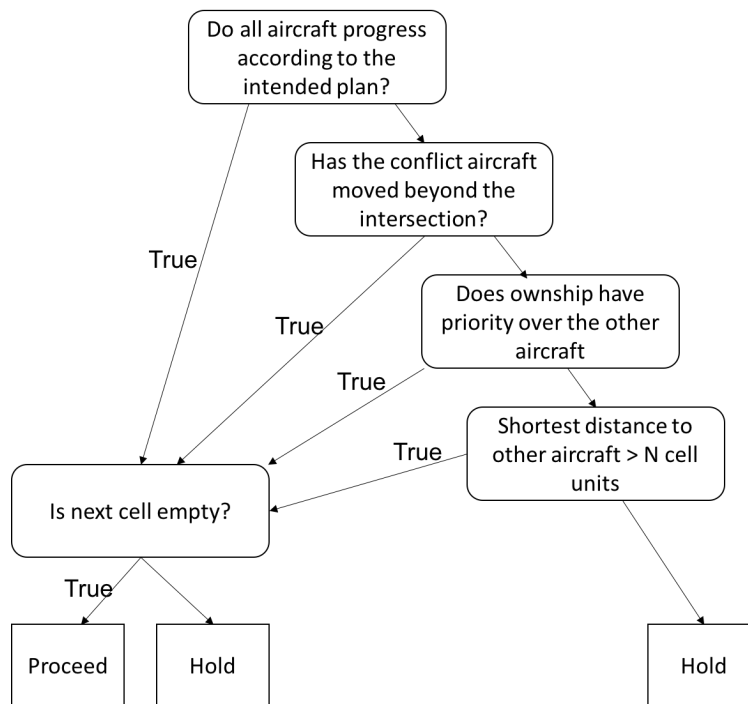


Figure 4: Decision tree to ensure crossing intersecting paths is safe

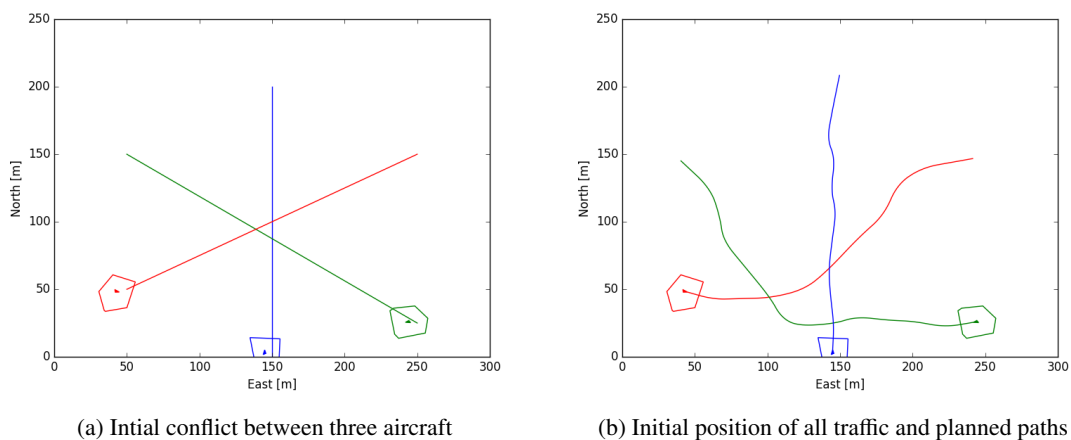


Figure 5: Execution of implicit coordination

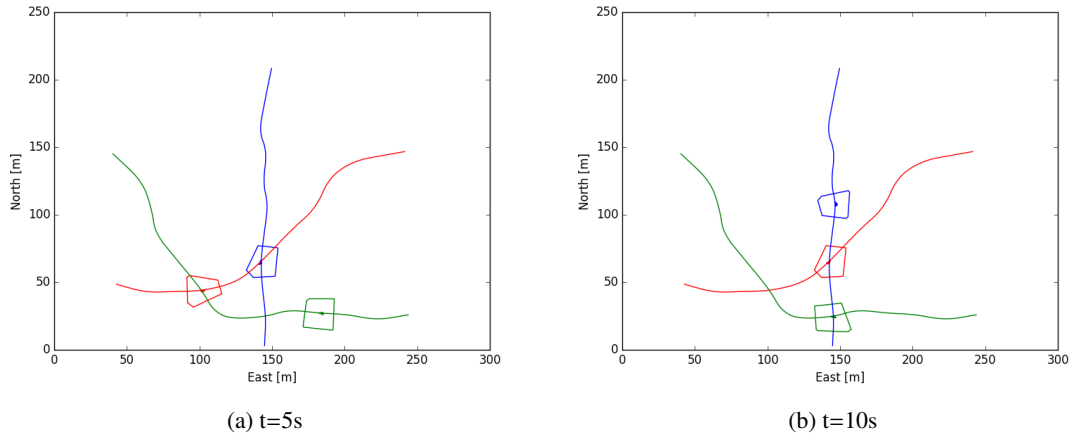


Figure 6: Execution of implicit coordination

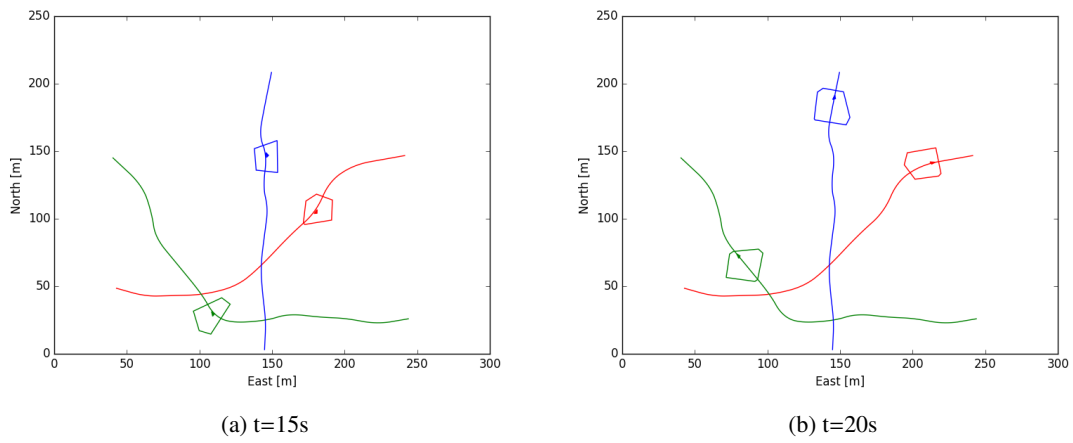


Figure 7: Execution of implicit coordination

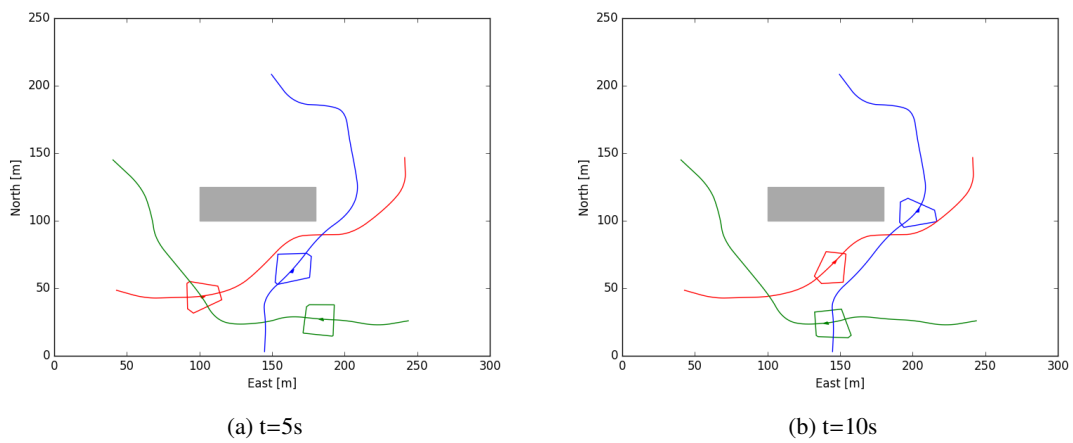


Figure 8: Execution of implicit coordination with a keep-out geofence

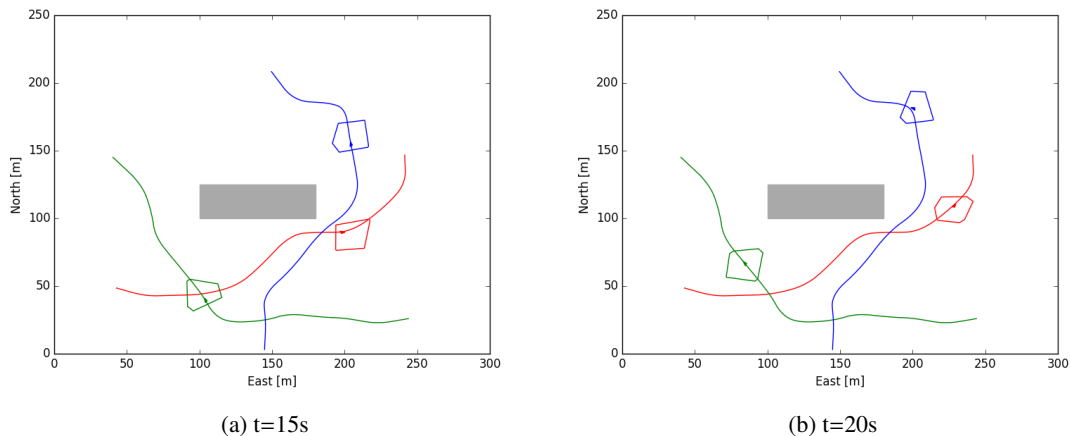


Figure 9: Execution of implicit coordination with a keep-out geofence

IV. Conclusions

This paper presents an approach to enable multiple aircraft to implicitly coordinate their resolution maneuvers. In the absence of a communication network, each aircraft involved in a conflict can generate resolution trajectories that are implicitly coordinated. This approach can also exploit the benefits of a communication network if available. In the presence of communication infrastructure, the approach can also be executed in a centralized fashion. Unlike other detect and avoid solutions, this approach enforces separation between aircraft by prescribing keep-in geofence constraints. Consequently, if all aircraft use a geofencing capability that is provably safe, it is possible to ensure that no two aircraft involved in a conflict will encounter a loss of separation with each other.

References

- ¹Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., and Robinson, J., "Unmanned Aircraft System Traffic Management (UTM) concept of operations," *16th AIAA Aviation Technology, Integration, and Operations Conference, AIAA Aviation*, 2016.
- ²Cook, S. P., Brooks, D., Cole, R., Hackenberg, D., and Raska, V., "Defining well clear for unmanned aircraft systems," *AIAA Infotech@Aerospace*, 2015, p. 0481.
- ³Kuchar, J. K. and Yang, L. C., "A review of conflict detection and resolution modeling methods," *IEEE Transactions on intelligent transportation systems*, Vol. 1, No. 4, 2000, pp. 179–189.
- ⁴Williamson, T. and Spencer, N. A., "Development and operation of the traffic alert and collision avoidance system (TCAS)," *Proceedings of the IEEE*, Vol. 77, No. 11, 1989, pp. 1735–1744.
- ⁵Kochenderfer, M. J., Holland, J. E., and Chrysanthacopoulos, J. P., "Next generation airborne collision avoidance system," *Lincoln Laboratory Journal*, Vol. 19, No. 1, 2012, pp. 17–33.
- ⁶Muñoz, C., Narkawicz, A., Hagen, G., Upchurch, J., Dutle, A., and Consiglio, M., "DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems," *Proceedings of the 34th Digital Avionics Systems Conference (DASC 2015)*, Prague, Czech Republic, September 2015.
- ⁷Asmar, D. M., *Airborne collision avoidance in mixed equipage environments*, Ph.D. thesis, Massachusetts Institute of Technology, 2013.
- ⁸Kosecka, J., Tomlin, C., Pappas, G., and Sastry, S., "Generation of conflict resolution maneuvers for air traffic management," *International Conference on Intelligent Robots and Systems*, Grenoble, 1997.
- ⁹Tomlin, C., Pappas, G., and Sastry, S., "Noncooperative Conflict Resolution," *IEEE International Conference on Decision and Control*, San Diego, CA, December 1997.
- ¹⁰Bicchi, A. and Pallottino, L., "On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000, pp. 221–232.
- ¹¹Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, August 2004.
- ¹²Granger, G., Durand, N., and Alliot, J.-M., "Token Allocation for Free-Flight Conflict Solving," *Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence Conference (IAAI-01)*, edited by H. Hirsch and S. Chien, American Association for Artificial Intelligence, Menlo Park, CA, Aug. 7–9 2001, pp. 59–64.
- ¹³Richards, A. and How, J. P., "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, Vol. 16, No. 7, 2006, pp. 333–351.
- ¹⁴Ong, H. Y. and Kochenderfer, M. J., "Short-term conflict resolution for unmanned aircraft traffic management," *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th*, IEEE, 2015, pp. 5A4–1.
- ¹⁵Bilimoria, K., "A Geometric Optimization Approach to Aircraft Conflict Resolution," *Guidance, Navigation, and Control Conference*, Vol. AIAA 2000-4265, Denver, CO, August 2000.

- ¹⁶Narkawicz, A. and Muñoz, C., “State-Based Implicit Coordination and Applications,” Technical Publication NASA/TP-2011-217067, NASA, Langley Research Center, Hampton VA 23681-2199, USA, March 2011.
- ¹⁷Dowek, G. and Muñoz, C., “Conflict Detection and Resolution for 1,2,...N Aircraft,” *Proceedings of the 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Belfast, Northern Ireland, September 2007.
- ¹⁸Zeghal, K., “Towards the logic of an airborne collision avoidance system which ensures coordination with multiple cooperative intruders,” *ICAS PROCEEDINGS*, Vol. 19, AMERICAN INST OF AERONAUTICS AND ASTRONAUTICS, 1994, pp. 2208–2208.
- ¹⁹Eby, M., “A Self-Organizational Approach for Resolving Air Traffic Conflicts,” *Lincoln Laboratory Journal*, Vol. 7, No. 2, 1994, pp. 239–254.
- ²⁰Hoekstra, J., “Designing for Safety: The Free Flight Air Traffic Management Concept,” Tech. Rep. 90-806343-2-8, Technische Universiteit Delft, Nov. 2001.
- ²¹Narkawicz, A. and Hagen, G. E., “Algorithms for collision detection between a point and a moving polygon, with applications to aircraft weather avoidance,” *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3598.
- ²²Owre, S., Rushby, J. M., and Shankar, N., “PVS: A prototype verification system,” *International Conference on Automated Deduction*, Springer, 1992, pp. 748–752.
- ²³LaValle, S. M. and Kuffner Jr, J. J., “Randomized kinodynamic planning,” *The international journal of robotics research*, Vol. 20, No. 5, 2001, pp. 378–400.
- ²⁴uAvionix, “uAvionix-Sense and Avoid for drones,” <http://www.uavionix.com/> accessed April, 2017.