https://ntrs.nasa.gov/search.jsp?R=20180005423 2019-04-30T05:09:41+00:00Z

Simulation Evaluations of an Autonomous Urban Air Mobility Network Management and Separation Service

Christabelle S. Bosson*

USRA/NAMS, NASA Ames Research Center, Moffett Field, CA 94035, USA

Todd A. Lauderdale†

NASA Ames Research Center, Moffett Field, CA 94035, USA

This paper presents an initial implementation of an autonomous Urban Air Mobility network management and aircraft separation service for urban airspace that does 1) departure and arrival scheduling across the network, 2) continuous trajectory management to ensure safe separation between aircraft, and 3) seamless integration with traditional operations. The highly-autonomous AutoResolver algorithm developed for traditional aviation was extended to provide these capabilities. An evaluation of this initial implementation was conducted in fast-time simulations using a dense, two-hour traffic scenario with Urban Air Mobility aircraft flying between a network of 20 vertiports in the Dallas-Fort Worth metroplex. When the spatial separation was reduced from 0.3nmi to 0.1nmi, the total delay decreased by 7.3%; when the temporal separation was reduced from 60s to 45s, the total delay decreased by 28.4%. The total number of conflict resolutions decreased by 26% and 17%, respectively. Furthermore, when a scheduling horizon greater than the duration of UAM flights was used (50min), most conflicts were resolved pre-departure producing ground delay. By comparison, when a shorter scheduling horizon was used (8min), most conflicts were resolved post-departure generating airborne delay. For all scheduling and separation constraints tested, AutoResolver prevented loss of separation from occurring. Urban Air Mobility operations have the ability to revolutionize how people and goods are transported and this paper presents initial research focusing on the high levels of autonomy required for an airspace system capable of scaling to handle significantly higher densities of aircraft.

I. Introduction

Over the past year, there has been an increased interest in Urban Air Mobility (UAM) operations integration around the world.[1,2] Recent technology improvements have leveraged the development of novel aircraft types and operations for which deployment would require changes in the way the airspace is accessed.[3–6] Enabling UAM in the National Airspace System (NAS) can revolutionize how people and goods are transported, including riding on-demand air taxis and delivering packages using autonomous aircraft.[7,8] These new types of missions imply operations will occur over urban areas, mixing with air traffic that is currently managed and separated by air traffic controllers. Some of the barriers to enabling UAM operations include vehicle systems certification, noise impacts from vehicle operations in urban zones, and cyber security protections. In this paper, we focus on a significant barrier: creating a safe airspace system to handle very-high-density operations while integrating with legacy airspace operations. The density of projected operations requires the integration of vehicles-piloted or not-into an air traffic control system that provides flexibility and equitable access to shared airspace resources. Enabling safe and efficient UAM operations in the NAS necessitates a large paradigm shift of the current air traffic control system towards higher levels of autonomy.

*Aerospace Engineer, Universities Space Research Association, NASA Ames Research Center, Moffett Field, CA 94035 AIAA Member.

†Aerospace Engineer, Aviation Systems Division, NASA Ames Research Center, Moffett Field, CA 94035 AIAA Member.

For many years, NASA has conducted research towards the development of algorithms to enable increased levels of integrated automation across the NAS.[9,10] In this paper, the Advanced Airspace Concept (AAC) for traffic in the enroute airspace and the Terminal Advanced Airspace Concept (TAAC) which extends AAC to terminal areas, serve as the basis for the conducted research.[9,11] AAC and TAAC form a set of concepts providing automated conflict detection and resolution, arrival and departure management, and weather avoidance as a basis for autonomous air traffic control. In a recent effort, AAC and TAAC were integrated into a single platform that ensures continuous functionalities across airspaces.[12] The architecture of the platform can be decomposed into two independent systems that provide conflict detection and resolution at different time frames.[13] The first component is responsible for guaranteeing tactical separation one minute or less to loss of separation. The second component, also known as the AutoResolver, is a trajectory based-algorithm that performs strategic separation looking out 20min for sequencing purposes and 1-10min for losses of separation. In past research, TAAC has been demonstrated to efficiently ensure separation assurance and arrival sequencing as well as departure management procedures in particular for traffic in the Dallas-Forth Worth D10 TRACON.[14] To date, the tool has been developed for traffic scenarios including commercial aircraft and business aviation.

In the research effort described in this paper, the AutoResolver algorithm has been extended to accommodate UAM-type vehicles and enable UAM operations in the NAS. This paper presents the exploration of the AutoResolver algorithm extended capabilities to safely and efficiently manage UAM aircraft operations. To help guide this exploration, a two-hour traffic scenario is created with electric, vertical takeoff and landing aircraft, scheduled at a rate of one every five minutes, and flying in a network of vertiports located in the Dallas-Fort Worth metroplex. The scenario is evaluated in fast-time simulations, and three algorithm evaluation conditions are compared to provide answers to the following objectives; (1) to understand how UAM operations can be enabled using the AutoResolver algorithm and what constraints need to be implemented in order to ensure conflict-free UAM operations; (2) to characterize the difficulties surrounding autonomously controlling UAM operations in dense and complex airspace and investigate different separation and sequencing standards between UAM-type vehicles; and (3) to understand the trade-offs between scheduling strategies for these operations.

The paper is organized as follows. The autonomous AutoResolver algorithm is presented in Section II. The problem statement and methodology are described in Section III. In Section IV, selected effects of network interactions are presented, and the results of the comparison of three algorithm evaluation conditions are detailed and analyzed in Section V. Section VI concludes the paper with a summary and directions for future work.

II. Autonomous AutoResolver Algorithm

The autonomous terminal airspace AutoResolver algorithm, commonly known as TAAC AutoResolver, is a trajectory-based algorithm that performs conflict detection and resolution in terminal areas.[11] Using the Airspace Concept Evaluation System (ACES) that provides realistic conflict scenarios (based on actual, flown aircraft routes) and trajectory predictions, AutoResolver was designed and developed. When running a simulation, AutoResolver is continuously called at each iteration time period, and ACES is used to fly the aircraft with high levels of accuracy. The inputs of the algorithm for each flight include the aircraft information, type and motion state, the departure and arrival airports, weather information as well as airspace configuration parameters. Figure 1 summarizes the algorithm functionalities in a functional diagram.

At each simulation time step, AutoResolver first performs conflict detection using four-dimensional (4D) aircraft projected trajectories computed by an external trajectory predictor module. The algorithm checks all aircraft for future scheduling conflicts and losses of separation. The conflicts are sorted by level of urgency based on time to loss of separation. Then, AutoResolver computes resolution maneuvers in a second step for all aircraft in detected conflicts. The algorithm uses a trial planning process to comprehensively solve both scheduling and loss of separation conflicts. Several resolution strategies are computed using speed control, altitude changes and horizontal maneuvers. The candidate resolution maneuvers, also called trial plans, are then sent out, one by one, to an external trajectory predictor which computes the aircraft full 4D trajectory from current position to landing. Each trial plan is then checked to validate that it solves the original conflict and doesn't introduce any secondary conflicts. The trial planning process does not stop after it finds the first successful resolution maneuver. Instead, it continues to search for additional resolutions and can therefore induce several successful maneuvers to be found. If multiple successful resolutions are computed,

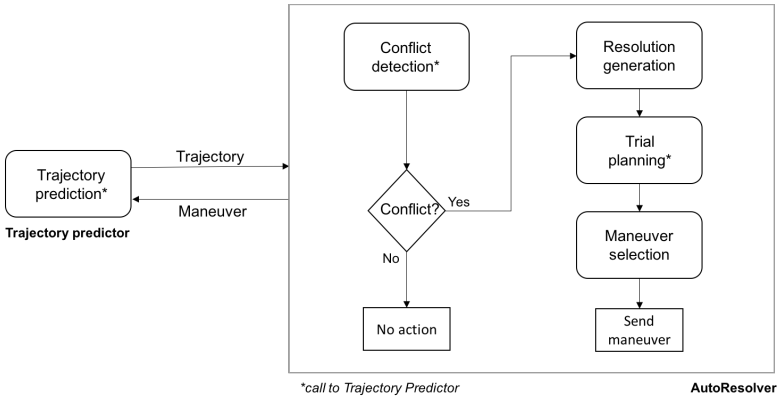


Figure 1: AutoResolver Functional Diagram

the algorithm then calls a selection module that, based on specified parameters to be optimized (commonly set as delay and fuel burn), selects the optimal resolution maneuver. The selected maneuver is then issued to the aircraft, and its trajectory is accordingly modified and updated. The described two-step process is periodically repeated at each simulation time step.

The AutoResolver algorithm has been applied in past research to traffic scenarios in the DFW metroplex containing medium and large transportation aircraft.[13,14] For the present work, AutoResolver's capabilities were adapted for UAM-type vehicles and missions by parametrizing AutoResolver's input variables and implementing UAM-dedicated software. Although the existing conflict detection logic was reused, a new resolution logic was designed and implemented to handle UAM aircraft and manage the operations. UAM-specific software was developed to compute resolutions in case of conflicts, and a catalog of maneuvers to choose from was built. Horizontal, vertical and speed maneuvering specific for electric, vertical takeoff and landing aircraft were designed and enabled. Examples of maneuvers include hovering and path stretching. The algorithm version that was used in this research was matured through the analysis of diverse simulation runs and parameter-tuning studies. As the research progresses, its implementation and feature set will continue to evolve to address questions related to energy and contingency planning topics, among others.

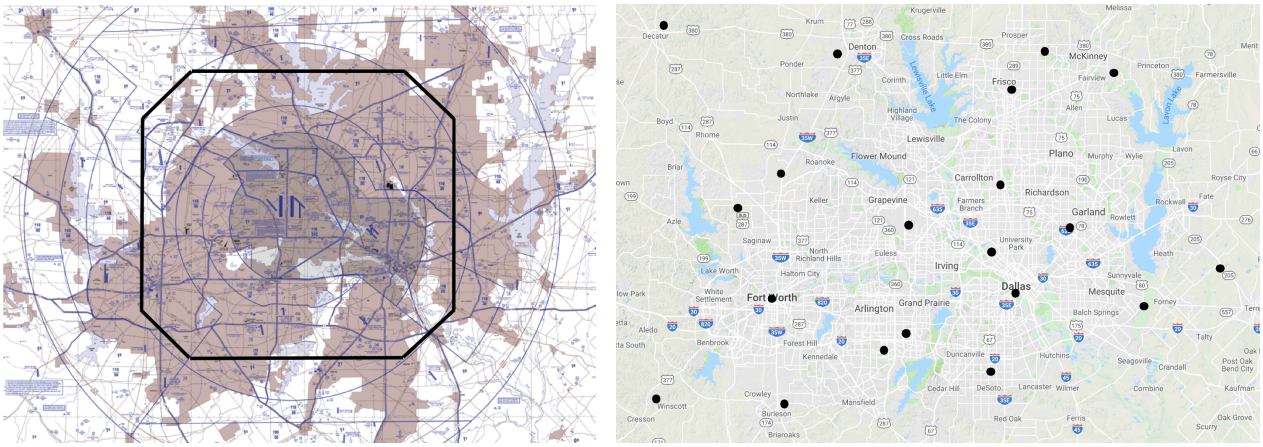
III. Methodology

To accomplish the goals established for this research, the AutoResolver algorithm is evaluated in fast-time simulations incorporating UAM aircraft. The airspace surrounding the Dallas Forth-Worth metroplex is selected, and a route network is constructed composed of 20 vertiports for the application. A network-wide schedule is generated using a takeoff rate of one every five minutes for all vertiports, and the resulting traffic scenario is used to evaluate the algorithm under different conditions, which are detailed in this section. The takeoff rate was chosen based on information from NASA's industry partners to enable the investigation of the major issues.

A. Airspace Details and Flight Route Network

The airspace surrounding the DFW metroplex is selected for the application and it is illustrated in Figure 2. The selected airspace represents a subset of the Terminal Radar Approach Control (TRACON) D10 which is a four corner post airspace containing 29 airports. The D10 TRACON is represented in Figure 2a, where currently traditional commercial operations are dominated by DFW airport followed by Dallas Love Field (DAL) airport activities. In the current concept, UAM operations can be distinguished from airline operations because they introduce highly autonomous aircraft with electric, vertical takeoff and landing capabilities. In this research, operations are assumed to initiate and end at various vertiport locations situated throughout the DFW metroplex and 20 existing heliports are selected and converted into vertiports. Figure 2b illustrates the geographic positions (black dots) of the potential vertiports, that are all located within the metroplex total area of $24,100\ km^2$.

Using the selected 20 vertiports, a route network, displayed in Figure 3, is constructed by connecting each

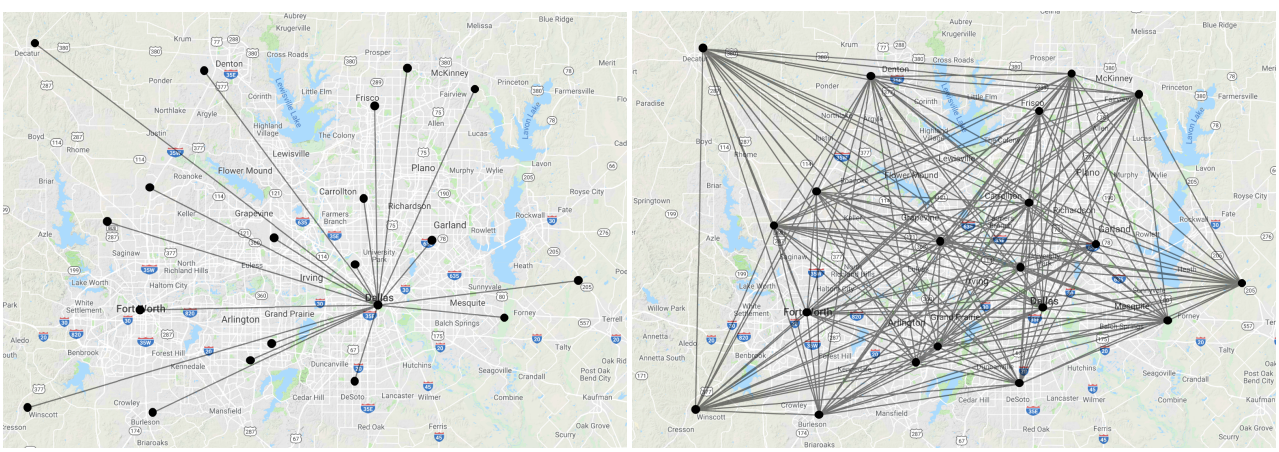


(a) D10 Tracon

(b) DFW Metroplex - 20 Vertiports

Figure 2: Airspace Details

vertiport to one another by straight lines for simplicity (and lack of existing routes). Intermediate waypoints are inserted along the route to create flight plans between each pair of vertiports. No flight departure and arrival procedures are modeled when leaving or arriving into a vertiport. The flight routes span distances ranging from 2 - 3nmi for the shortest to 30nmi for the longest. Figure 3a illustrates the route connectivity from the Dallas downtown vertiport to the other 19 vertiports. Figure 3a shows the tightness of the route structure linking the vertiports. The route network complexity is furthermore exposed in Figure 3b where it is almost impossible to visually distinguish all 190 distinct routes.



(a) Dallas Downtown Vertiport Route Connectivity

(b) System Wide Flight Route Network

Figure 3: Flight Route Network Structure

The selected vertiports are located at various places such as municipal airports, hospital helipads, roofs of buildings, and so on. Therefore, the vertiports might not necessarily be on the ground. The altitudes of the considered vertiports range between 480ft and 1047ft Mean Sea Level (MSL). The altitude of the DFW airport is 607ft MSL for reference. The route network assigns flights to altitude levels ranging between 800ft MSL and 1600ft MSL to adapt to the different route lengths. This ensures the cruising altitude to always be above the altitude of the landing vertiport. Both directions of each distinct route are implemented, thus the size of the route set is 380. The flight route network constructed for this application is dense, both horizontally and vertically.

B. Network-Wide Traffic Schedule

A network-wide schedule creating two hours of UAM traffic is generated for the route structure previously described. A rate of one takeoff every five minutes is imposed at all 20 proposed vertiports. The takeoff rate was chosen based on information from NASA's industry partners. For simplicity, arrival and departure operations are assumed to occur at a single pad at each vertiport, and activities such as passenger unloading/loading, vehicle recharging and maintenance are not modeled, but are assumed to always be completed prior to the next 5min takeoff window. Moreover, since only one pad is modeled per vertiport, an aircraft can depart immediately after an aircraft lands, if no loss of separation with surrounding aircraft is detected. Due to the nature of on-demand mobility services, the schedule generation incorporates some randomness in the selection of the destination vertiport. For each departure vertiport and each takeoff time window, spaced every five minutes, the destination is randomly selected from the other 19 vertiports. To illustrate the generated schedule, Figure 4 represents the takeoff timeline for several vertiports as well as for the entire network. Each colored tick denotes a takeoff time expressed in minutes and seconds. In two hours of traffic, there are 24 takeoff time windows and each takeoff time window contains 20 takeoff times. To prevent departures from being exactly synchronized at each vertiport, the takeoff time values were assigned using a distribution for which the samples are uniformly distributed over the interval $[0, 10]$s. The network-wide traffic schedule was created at first by randomly assigning the destination, without the consideration of any altitude nor speed routing constraints. Thus, infeasible routes and schedules were generated such as routes of length less than 2nmi and schedules assigning aircraft to fly head-on at the same altitude between any two vertiports at close time periods. To rule out these infeasible routing and scheduling options, distance and altitude checks were then made. In Figure 4, takeoff times are missing in some takeoff time windows, such as in the last few for vertiport 2, because they corresponded to infeasible routes and thus filtered out from the schedule. There is no specific reason why infeasible routes were not generated in the first 80 minutes, the route destinations were randomly assigned.

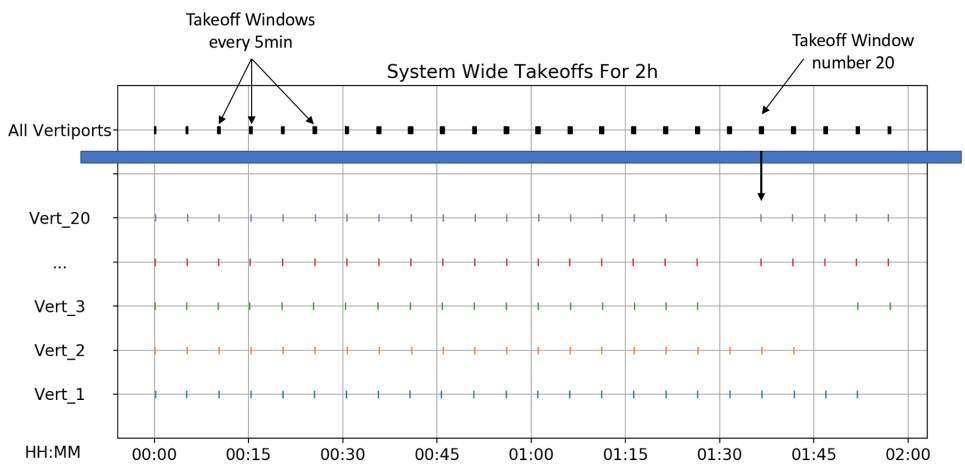


Figure 4: Network Wide Traffic Takeoff Timeline

C. Aircraft Characteristics and Trajectory Generation

UAM-type vehicles with electric propulsion and vertical takeoff and landing capabilities (i.e. eVToL) are currently being designed, manufactured and tested by various companies around the world.[3–6] Given the emerging nature of UAM missions and early stage of aircraft development, no common eVToL aircraft models exist. Although various vehicle designs, investigating different propulsion systems, are being developed, they all share a common goal: enable everyday aerial point-to-point transportation. Thus, a simplified kinematic linear interpolation-based trajectory generator (ITG) is implemented for this research that allows basic horizontal and vertical maneuvering. The flight phases are decomposed between vertical takeoff, climb, cruise, descent and vertical landing. It is assumed that a flight-transition occurs between takeoff and climb, and between descent and landing while the aircraft hovers. Except for the vertical takeoff and landing

phases, UAM eVToL aircraft characteristics are derived based on the Cessna 172 aircraft performance flying qualities. The reference capabilities are extended to a climb rate of 800fpm and an airspeed of 170kts during cruise for the UAM aircraft. Figure 5 illustrates a generated trajectory using the ITG between two vertiports cruising at 1500ft. Several types of maneuvers are implemented using ITG allowing horizontal, vertical and speed maneuvering.

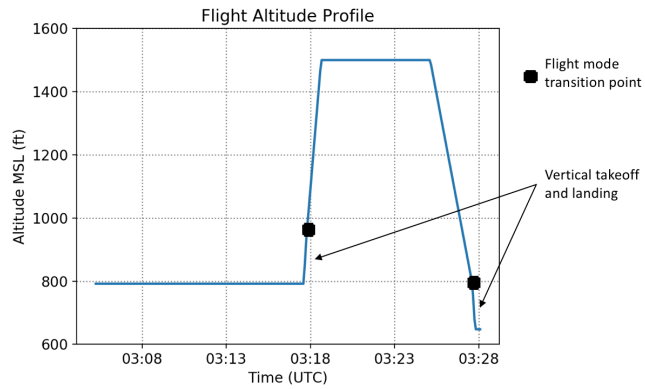


Figure 5: UAM eVTol Trajectory Representation

D. Evaluation Conditions

Three independent variables were chosen to explore the effects of integrating UAM aircraft into the airspace using the AutoResolver algorithm and investigate different management policies: separation requirements, sequencing specifications and arrival scheduling strategies. After evaluating the AutoResolver algorithm on the described route network and flight schedule without mitigations, the algorithm resolution mode was enabled and the evaluation of eight different algorithm conditions was conducted through simulations. Table 1 presents the test matrix. The separation standards impose horizontal and vertical separations between aircraft, whereas the sequencing specification imposes temporal separation between aircraft. The arrival scheduling horizon specifies when the aircraft are scheduled to their arrival vertiports.

Table 1: Test Matrix Variable Description

Test Case	Separation Standards	Sequencing Specification	Arrival Scheduling Horizon
1. Baseline	0.3nmi H 100ft V	60s	50min
2. Reduced lateral spacing	0.1nmi H 100ft V	60s	50min
3. Reduced temporal spacing	0.3nmi H 100ft V	45s	50min
4. Reduced arrival scheduling horizon	0.3nmi H 100ft V	60s	8min

The separation standards were chosen using the definition of well clear for Unmanned Aircraft Systems (UAS) according to Cook and Brooks.[15] For large UAS in high-altitude airspace, the Phase 1 Detect And Avoid (DAA) well clear definition states the Horizontal Miss Distance (HMD) to be 0.66nmi, the Vertical Miss Distance (VMD) to be 450ft, and the modified τ to be 35s using a Distance Modification (DMOD) of 0.66nmi. For small UAS (55lbs vehicle or less) in low-altitude controlled airspace around airports, Wu et al. selected the horizontal and vertical separations, respectively to be a HMD of 0.36nmi and a VMD of 450ft, and a modified τ to be 15s.[16] Using those values as reference, the nominal spatial separation standards picked for this UAM application are set to 0.3nmi horizontally and 100ft vertically. These are tighter than for UAS because it is assumed that enhanced equipage capabilities will be installed on board UAM aircraft. To further assess the impact of reduced separations for this application, horizontal separation standards are then reduced to 0.1nmi, in a second step, close to the Near mid-air collision (NMAC) standard as defined by the Aeronautical Information Manual (7-6-3).[17] For the sequencing specification, a temporal separation

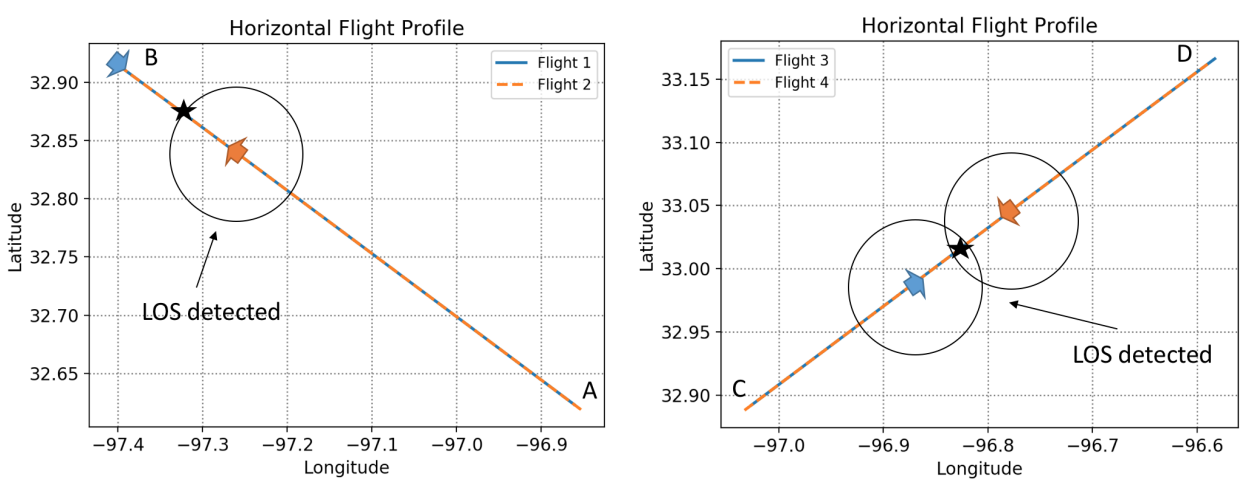
of 60s is first imposed to control the arrival flow rate at each vertiport. This value is inspired from the traditional wake vortex separations imposed by the Federal Aviation Administration (FAA) between aircraft of category F landing in sequence on the same runway.[18] The temporal separation is then reduced to 45s to exercise further AutoResolver arrival management capability. Moreover, it is envisioned in this preliminary phase of UAM research and development that UAM flights will be scheduled prior to departure and allowed to take off if and only if the destination vertiport is projected to be open at the time of arrival. With this in mind, an arrival scheduling horizon of 50min is first set, forcing all flights to be scheduled before taking off. The arrival scheduling horizon is then reduced to 8min to understand the impacts of scheduling while in-flight and the trade-offs between the two different arrival scheduling strategies.

IV. Effects of System-Wide Network Interactions

In order to evaluate the algorithm under the previously described conditions, the AutoResolver algorithm capabilities have been extended to enable UAM operations management. Throughout the development process, challenges had to be overcome, especially in the design of the scenario. Three interesting challenges, detailed in this section, were selected as unique to UAM missions because they highlight difficulties in generating a flexible network and they significantly differ from traditional aviation air traffic management considerations. Understanding the network effects is crucial to design the correct set of airspace constraints to enable successful conflict-free UAM operations. The scenario was intentionally built with a minimum set of operational constraints to allow the exploration of an unconstrained network. Throughout the scenario development process, routing rules had to be implemented in order to remove unnecessary pressure on the algorithm conflict resolution process.

A. Colinear Flight Plan Trajectories

The designed flight route network contains routes that connect bi-directionally each pair of vertiports. Due to the randomness introduced in the schedule generation, some flights are scheduled to fly head-on at the same altitude between any two vertiports at close time periods. Because no altitude-for-direction rules were imposed to prevent this condition, losses of separation (LOS) are created between the initial trajectories that are colinear to one another. If this happens too often, unnecessary delays would be generated both on the ground and in the air. Figure 6 illustrates two examples of colinear trajectories connecting vertiport pairs A and B (Figure 6a), and C and D (Figure 6b).



(a) LOS Close to Vertiport Area

(b) LOS in Cruise

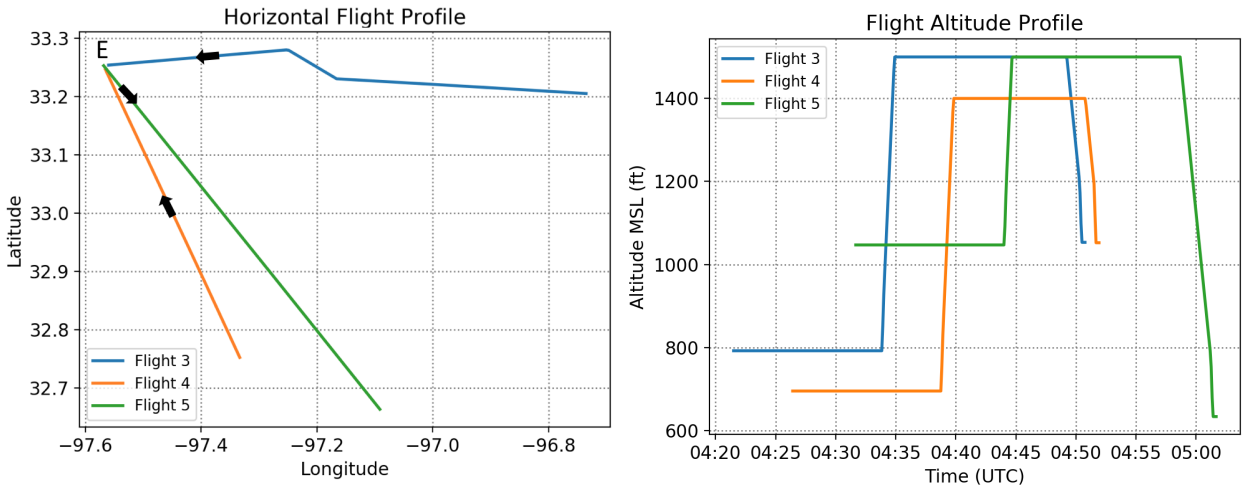
Figure 6: Colinear Flight Trajectories

In one example in Figure 6a, the LOS is detected close to vertiport B between the arriving Flight 2 and the departing Flight 1. In this situation, the LOS is solved by delaying the departing aircraft, Flight 1 on the ground. In the other example in Figure 6b, the loss of separation is detected in cruise for both Flight 3 and

Flight 4. In this situation, the LOS is solved by path stretching the trajectory of Flight 3. Both cases generate some delay which could have been partially avoided if altitude-for-direction rules had been considered. This example provides an illustration of some of the effects of system-wide network interactions, and shows that some outcomes could be mitigated by altitude layering considerations in the routing generation to avoid unnecessary delays.

B. Routing Between Vertiports of Different Altitudes

The designed flight route network contains routes that connect vertiports located at different altitudes. Thus, some flights might have route portions superimposed in altitude, some might start at a lower altitude than the one at which they end and vice versa. Figure 7 illustrates a flight scenario for which three aircraft depart from and arrive at vertiports located at different altitudes.



(a) Horizontal Trajectories

(b) Vertical Trajectories

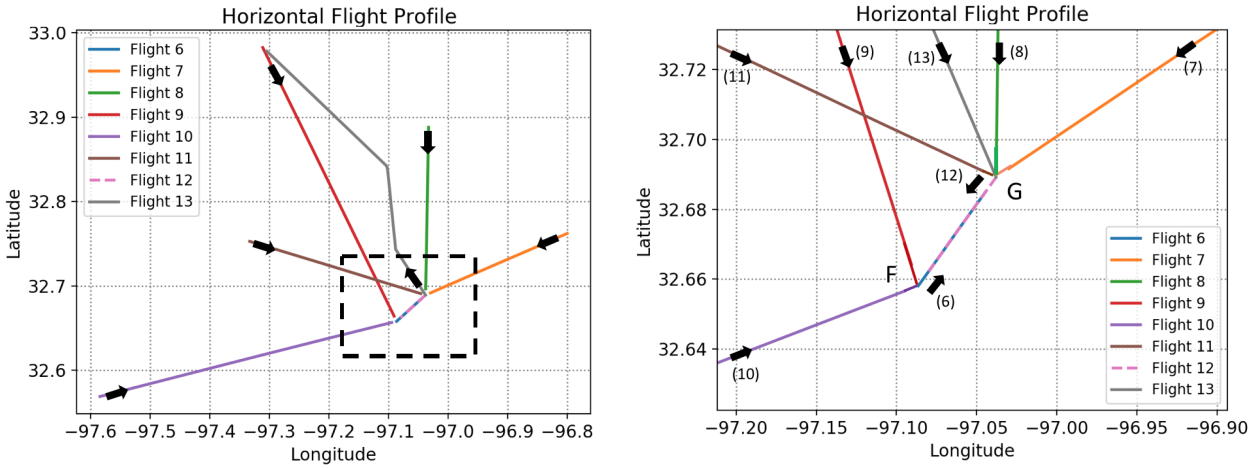
Figure 7: Flight Routes Between Vertiports of Different Heights

Figure 7a shows that the three flights have vertiport E in common, and the trajectories do not present any colinearity characteristics. Figure 7b displays that Flight 3 and Flight 4 land on vertiport E located at 1047 ft MSL, whereas Flight 5 departs from it. From the figure, it can be seen that Flight 3 and Flight 4 depart from two separate vertiports that are respectively located at 795 ft MSL and 690ft MSL. In time order, Flight 5 takes off from vertiport E, Flight 3 lands, then it is followed by Flight 4. Figure 7b demonstrates that the trajectories of Flight 3 and Flight 4 are superimposed in altitude after some time in cruise, and because of the altitude of the landing vertiport, both flights have a short descent. Because Flight 4 flies a short route and to prevent any sequencing conflict at vertiport E, its cruising altitude was set to 1400ft MSL. Such example highlights the difficulties in generating flight plans and schedules for a complex and dense network. In particular, if the network has vertiports of different altitudes, the assigned cruise altitude by the scenario, needs to be high enough to allow enough control authority in descent for spacing and sequencing. This creates a challenge that is made more difficult for short flights.

C. Interdependent Vertiports

The designed flight route network contains routes between vertiports that are as close as 2nmi apart. Figure 8 depicts an eight-flight scenario for which the operations are concentrated around vertiports F and G. The two vertiports are located 4.3nmi apart.

Figure 8a illustrates the horizontal trajectories of the considered flights, and Figure 8b offers a zoom in on the trajectories in the area surrounding vertiports F and G. First, it can be observed that Flight 6 departs vertiport F whereas Flights 9, 10 and 12 arrive into vertiport F. Second, it can be seen that Flights 12 and 13 depart vertiport G whereas Flights 6, 7, 8 and 11 arrive into vertiport G. Flights 6 and 12 operate between the two vertiports. This scenario shows that scheduling operations out of vertiports F and G are not



(a) Horizontal Trajectories

(b) Zoomed Horizontal Trajectories

Figure 8: Flight Routes Linking Interdependent Vertiports

independent of one another. Their proximity implies that if the operations are not correctly synchronized between the two locations, consequences leading to increased delay that impact the whole system will occur. This example highlights coupling effects that could be managed by creating shared arrival and departure procedures for vertiplex areas.

V. Simulation Results

Fast-time simulations were conducted to examine the extended UAM capabilities of the AutoResolver algorithm. This section presents the results. High-level metrics including delays and number of resolutions were computed for all runs. No trajectory uncertainty was modeled in this research.

A. AutoResolver Detection-Only Results

Before enabling the resolution capabilities of the AutoResolver algorithm, the generated network-wide schedule was evaluated without mitigating potential conflicts. The number of unique Losses of Separation (LOS) between each aircraft pair is reported in Table 2 for each previously described separation standard. The computed values are shown to highlight the complexity and density of the generated routing and schedule network.

Table 2: Results - AutoResolver Simulations Detection Only

Separation Standards	Number of Unique LOS Between Aircraft Pairs
0.3nmi H 100ft V	36
0.1nmi H 100ft V	9

Without mitigations, 36 losses of separation occurred when the horizontal separation-standard was set to 0.3nmi, whereas 9 losses of separation occurred when the horizontal separation standard was set to 0.1nmi. The observed decrease in losses of separation was expected because when the horizontal separation-standard was reduced, aircraft were allowed to fly closer to one another. Therefore, fewer separation conflicts were generated.

B. AutoResolver Detection and Resolution Results

The results of the simulations with both detection and resolution enabled are detailed in the following subsections. Four evaluation conditions are selected to highlight the results; one is used as baseline (referred as test case 1 in Table 1), and the three others are used to demonstrate the respective effects of reducing the values of the three decision variables (referred as test cases $2-4$ in Table 1). In the three following subsections, one simulation per reduced decision variable, respectively spatial, temporal and arrival scheduling as shown in Table 1, is individually compared to the baseline run using metrics based on delay and on the number of resolutions. The baseline simulation is the same in all three comparisons. For all simulations, AutoResolver was able to solve all losses of separation and sequencing conflicts without creating new ones. Before diving into the test case comparisons, the resulting number of airborne aircraft at any given time is displayed in Figure 9 for the Baseline run.

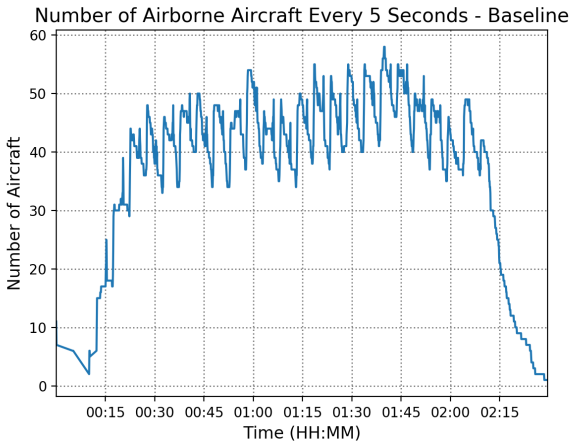
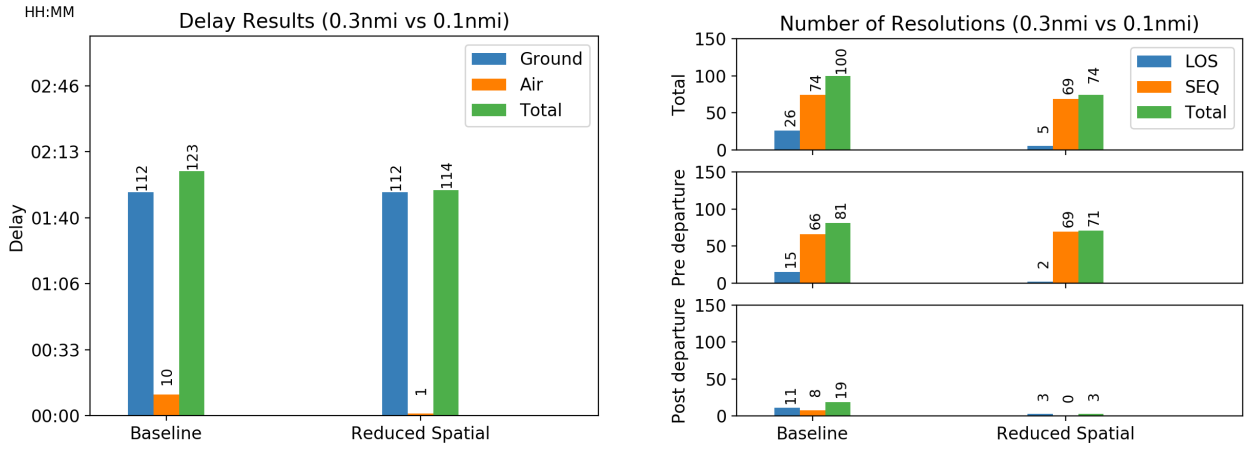


Figure 9: Number of Airborne UAM Aircraft at Any Given Time - Baseline

1. *Separation Standards Comparison (0.3nmi vs 0.1nmi)*

First, two separation-standard strategies were evaluated, 0.3nmi versus 0.1nmi, and the results are presented in Figure 10 as Baseline versus Reduced Spatial. The two simulations differed in horizontal separation but not in temporal separation nor arrival scheduling.

First, as illustrated in Figure 10a, when the horizontal separation was decreased from 0.3nmi to 0.1nmi, it was observed that the airborne delay decreased from 10s to 1s, whereas the ground delay remained constant and equal to 112s. In reduced spatial separation conditions, aircraft were allowed to spatially fly closer to one another. Therefore, it was observed that less airborne time was required to spatially separate the aircraft, which induced a decrease of airborne delay. On the other hand, it was observed that the spatial separation did not impact the ground operations, and ground delay wasn't affected between the two evaluations. Furthermore, because the reduction of horizontal separation allowed aircraft to fly closer to one another, a reduction of the total number of conflict resolutions was observed, from 100 to 74 respectively shown in Figure 10b, representing a decrease from 74 to 69 for sequencing conflicts and from 26 to 5 for losses of separation. The impact of reducing the spatial separation from 0.3nmi to 0.1nmi was also observed when looking at conflict resolution counts pre and post-departure. The number of resolutions for losses of separation decreased both pre and post-departure. This behavior shifted to increasing the number of sequencing conflicts to be solved pre-departure from 66 to 69, since aircraft presented fewer spatial separation issues. However, a decrease was observed post-departure from 15 to 2. When a smaller spatial separation was authorized between aircraft, more sequencing conflicts needed to be solved pre-departure because both spatial and temporal separations were imposed all the way down to the vertiports. The results obtained when reducing the separation standards were as expected and showcased the ability for AutoResolver to continuously ensure safe separation between aircraft in complex scenarios presenting both spatial and temporal constraints.



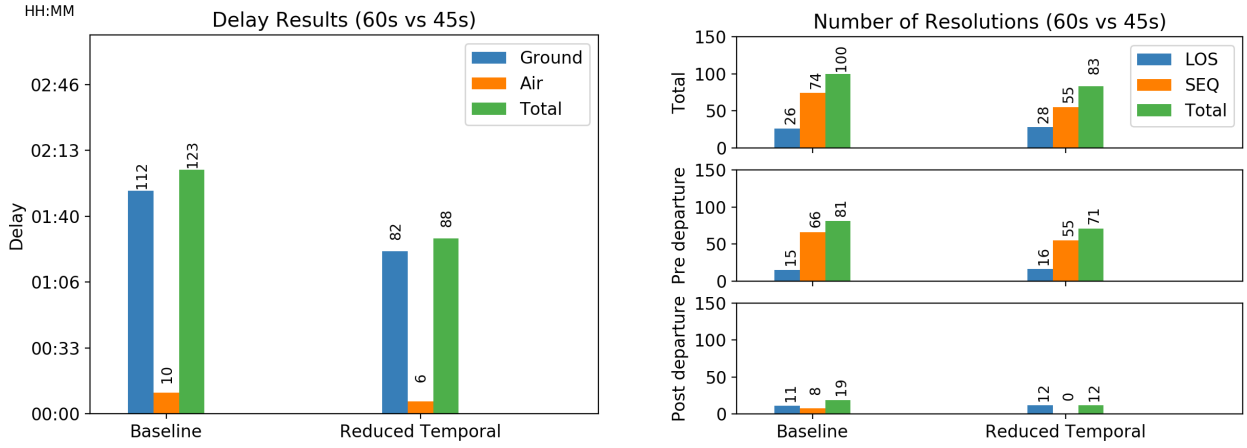
(a) Simulation Delays

(b) AutoResolver Executed Resolutions

Figure 10: AutoResolver Simulation Detection and Resolution Results: Baseline (0.3nmi H, 100ft V, 60s, 50min) vs Reduced Spatial Separation (0.1nmi H, 100ft V, 60s, 50min)

2. *Sequencing Requirements Comparison (60s vs 45s)*

Second, two sequencing requirement strategies were evaluated, 60s versus 45s, and the results are depicted in Figure 11 as Baseline versus Reduced Temporal. The two simulations differed in temporal separation but not in spatial separation nor arrival scheduling.



(a) Simulation Delays

(b) AutoResolver Executed Resolutions

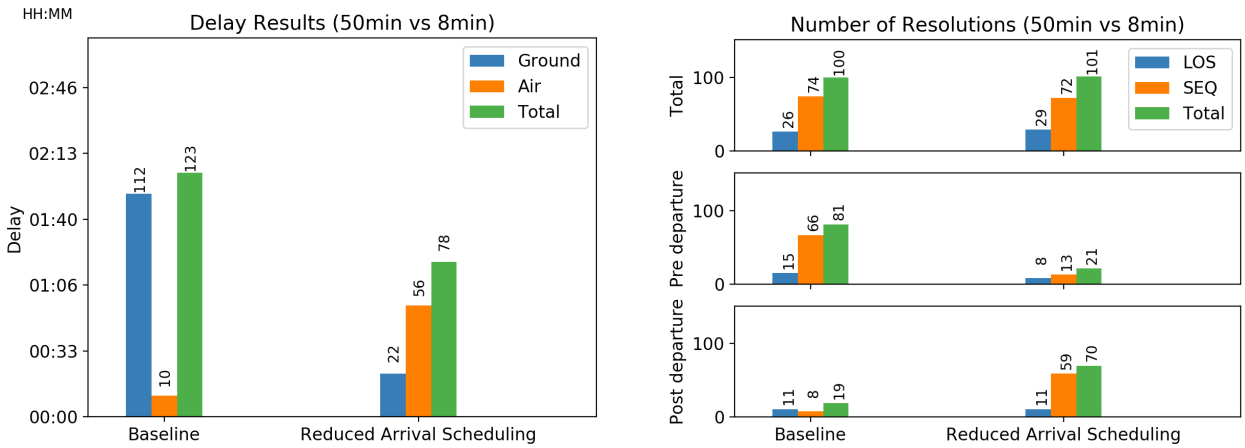
Figure 11: AutoResolver Simulation Detection and Resolution Results: Baseline (0.3nmi H, 100ft V, 60s, 50min) vs Reduced Temporal Separation (0.3nmi H, 100ft V, 45s, 50min)

It can be observed in Figure 11a that when the temporal spacing requirement was decreased from 60s to 45s, it induced the total delay values to decrease from 123s to 88s. This represents a larger reduction than was observed under reduced spatial standards. Specifically, the airborne delay decreased from 10s to 6s and the ground delay decreased from 112s to 82s, respectively. The reduced temporal requirements allowed less delay to be generated on the ground when scheduling the aircraft because less time was imposed between landing aircraft at the vertiports. In terms of conflict resolution numbers, a reduction of the total number of resolutions from 100 to 83 can be observed in Figure 11b for 60s vs 45s temporal separation. However this

represents a smaller total reduction than the one induced by reduced spatial separation-standards because the temporal separation only affected the aircraft sequencing and scheduling at the vertiports. The reduction in number of resolutions can be observed both pre and post-departure. Because all aircraft were scheduled prior to taking off, it was observed that smaller temporal separation induced the total number of resolutions computed pre-departure to decrease from 81 to 71. The number of resolutions for sequencing conflicts decreased from 66 to 55 whereas the number of resolutions for losses of separation increased from 15 to 16. It was also observed that reduced temporal requirements had less impact on the operations post-departure because all schedules were computed pre-departure, however a decrease in total number of resolutions can be observed from 19 to 12. The number of resolutions for sequencing conflicts decreased from 8 to 0, whereas the number of resolutions for losses of separation increased from 11 to 12. The small increase in the number of losses of separation both pre and post-departure is a consequence of allowing aircraft to be temporally closer to nearby arrival vertiports. The results of this study were as expected and demonstrated that the arrival management capability of AutoResolver was able to handle reduced temporal separations and sequence aircraft temporally closer to one another without increasing the number of losses of separation.

3. Arrival Scheduling Horizon Comparison (50min vs 8min)

Lastly, two arrival scheduling freeze horizons, 50min versus 8min, were evaluated, and the results are shown in Figure 12 as Baseline versus Reduced Arrival Scheduling. The two simulations differed in arrival scheduling horizons but not in spatial nor temporal separation.



(a) Simulation Delays

(b) AutoResolver Executed Resolutions

Figure 12: AutoResolver Simulation Detection and Resolution Results: Baseline (0.3nmi H, 100ft V, 60s, 50min) vs Reduced Arrival Scheduling Horizon (0.3nmi H, 100ft V, 60s, 8min)

When the arrival scheduling horizon value was decreased from 50min to 8min, it was observed that aircraft with flight times longer than 8min, were scheduled in-flight after taking off. Those with flight times less than 8min, remained scheduled on the ground before taking off. It can be observed in Figure 12a that reducing the arrival scheduling horizon value from 50min to 8min induced the total delay to decrease from 123s to 78s, a decrease larger than under the two previous reduced conditions. By allowing some aircraft to take off prior to being scheduled, less ground delay was generated and a decrease from 112s to 22s was observed. However, this shifted portions of ground delay to airborne delay, which was observed to increase from 10s to 56s. In contrast to the two previous reduced conditions, the decrease in arrival scheduling horizon didn't reduce the number of resolutions, and this can be observed in Figure 12b. The aggregate values of resolution numbers for losses of separation conflicts and sequencing conflicts are almost the same plus or minus two resolutions. However, when looking at the resolution numbers pre and post-departure, the previously observed trend is observed to be inverted between pre and post-departure. When using a smaller arrival scheduling horizon, it can be observed that scheduling operations were less concentrated on pre-departure and shifted to post-departure flight phase. After ensuring that no new losses of separation

would be created, many aircraft were released for departure prior to being scheduled. This explains the observed reduction of resolution numbers pre-departure from 15 to 8 for loss-of-separation conflicts and from 66 to 13 for sequencing conflicts. Then, schedules were computed when aircraft were in-flight, based on vertiport availability, and sequencing conflicts were resolved in order to find slots in the arrival stream. This is observed to not affect the number of losses of separation post-departure. Figure 12b displays that the number of resolutions for sequencing conflicts post-departure increased from 8 to 59. This last study illustrated that not only did AutoResolver continuously ensure the required spatial and temporal separation between all aircraft, it also presented efficient active scheduling control capabilities.

VI. Conclusion and Future Work

A shift from current human-centric air traffic control systems towards higher levels of autonomy is required to safely and efficiently integrate Urban Air Mobility operations in the National Airspace System. Towards that end, NASA's highly autonomous AutoResolver algorithm, with its continuous separation, departure and arrival management capabilities, was extended to Urban Air Mobility aircraft and missions. A system-wide network including routing and scheduling to and from vertiports was created for fast-time simulations of Urban Air Mobility operations in the Dallas-Fort Worth metroplex. A two-hour traffic scenario was built with electric, vertical takeoff and landing eVToL aircraft scheduled at a rate of one every five minutes among 20 vertiports.

After highlighting interesting trajectory insights induced by the effects of network interactions that were faced during the algorithm development, algorithm evaluation results of three different conditions were presented. Three decision variables – the separation standards, the sequencing specification and the arrival scheduling horizon – were selected and varied to help characterize the difficulties involved in autonomously controlling aircraft operations in dense and complex airspace. First, a baseline run was performed for reference with separation standards of 0.3nmi horizontally and 100ft vertically, a sequencing requirement of 60s, and an arrival scheduling horizon of 50min. Then three simulations were run to demonstrate the respective effects of reducing the values of the three decision variables. Each corresponding evaluation condition was individually compared to the baseline run using metrics based on delay and number of resolutions. In all simulations, AutoResolver was able to resolve all losses of separation and sequencing conflicts without creating new ones.

First, the evaluation of reduced spatial separation standards demonstrated that when horizontal spatial spacing was reduced to 0.1nmi, less delay was observed both on the ground and in the air, and fewer conflict resolutions were needed. Second, when reduced temporal separation requirements were reduced from 60s to 45s, a larger decrease in delay, both air and ground, and in the number of conflict resolutions was observed. Lastly, the evaluation of reduced arrival scheduling horizon conditions exposed the network effects of in-flight scheduling. When the arrival scheduling horizon was reduced to 8min, a larger decrease in total delay was observed, with a shift from ground to airborne delay. Moreover, the results showed that when the scheduling horizon was reduced, more conflicts needed to be resolved post-departure than pre-departure, inverting the trend observed in the baseline run. The algorithm evaluations of three different conditions demonstrated the capabilities of the AutoResolver algorithm at efficiently solving both sequencing and loss-of-separation conflicts regardless of the values of the decision variables. Although setting a large arrival scheduling horizon offered a conservative approach where only a few aircraft had to be maneuvered in flight, it induced significant ground delays. When aircraft were allowed to takeoff before being scheduled, less delay was generated for the network, and AutoResolver successfully computed solutions.

This paper demonstrates that AutoResolver is a candidate software with the potential to tackle the integration of Urban Air Mobility aircraft and operations in the National Airspace System. Its current capabilities not only enable the scheduling and sequencing of aircraft at vertiports, they enable active control of trajectories and can maneuver aircraft as needed. The AutoResolver algorithm can be used as a service that manages the entire Urban Air Mobility network with both strategic and tactical scheduling capabilities.

Future work is required to continue expanding the AutoResolver algorithm capabilities and to help answering Urban Air Mobility airspace integration research questions. First, the flight route network needs improvement. It will incorporate existing helicopter routes of the Dallas Forth-Worth metroplex, and obstacle heights will be shaped so that buildings are avoided when maneuvering aircraft. Vertiport areas with multiple vertipads need to be modeled, and their operations' management logic needs to be implemented. Trade-off studies will be carried out to compare different vertiport/vertipad configurations. Second, the traf-

fic scenario needs to be expanded to consider other aircraft types including aircraft flying Instrument Flight Rules and Visual Flight Rules, and interactions between mixed aircraft types need to be analyzed. Third, different demand sets could be generated that capture the demand preference of certain routes over others, and different takeoff rates could also be investigated. A separate network-wide scheduler can potentially be built to feed the AutoResolver algorithm through a web interface. Finally, parametric studies will be conducted to tune the algorithm parameters as part of evaluations with trajectory uncertainties modeled. New AutoResolver features are required to be developed to incorporate aircraft energy modeling and contingency management planning. These enhancements to the AutoResolver algorithm will allow further investigations of Urban Air Mobility airspace integration.

References

[1]Uber Elevate, "Fast-Forwarding to a Future of On-Demand Urban Air Transportation," 2017, `https://www.uber.com/elevate.pdf` (accessed November 20, 2017).

[2]Volocopter, "Reinventing Urban Mobility," 2017, `https://www.volocopter.com/en/` (accessed November 20, 2017).

[3]Aurora, "eVTOL," 2017, `http://www.aurora.aero/evtol/` (accessed November 20, 2017).

[4]Joby Aviation S2, "S2," 2017, `http://www.jobyaviation.com/S2/` (accessed November 20, 2017).

[5]A3 by Airbus Group, "Vahana," 2017, `http://www.airbus-sv.com/projects/1` (accessed April 13, 2018).

[6]Zee Aero, "Cora," 2017, `http://www.cora.aero` (accessed April 13, 2018).

[7]Airbus, "Rethinking Urban Air Mobility," 2017, `http://www.airbus.com/newsroom/topics-in-focus/urban-air-mobility.html` (accessed November 20, 2017).

[8]Amazon Prime Air, "First Prime Air Delivery," 2017, `https://www.amazon.com/Amazon-Prime-\%20Air/b?node=8037720011` (accessed November 20, 2017).

[9]Erzberger, H., Lauderdale, T., and Chu, Y., "Automated Conflict Resolution, Arrival Management, and Weather Avoidance for Air Traffic Management," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of aerospace engineering*, Vol. 226, No. 8, 2012, pp. 930–949.

[10]Guerreiro, N. M., Jones, D. R., Barmore, B. E., Butler, R. W., Hagen, G. E., Maddalon, J. M., Ahmad, N. N., Rogers, L. J., Underwood, M. C., and Johnson, S. C., "An Advanced Trajectory-Based Operations Prototype Tool and Focus Group Evaluation," *NASA TM2017-219670*, 2017.

[11]Erzberger, H., Nikoleris, T., Paielli, R. A., and Chu, Y.-C., "Algorithms for Control of Arrival and Departure Traffic in Terminal Airspace," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 230, No. 9, 2016, pp. 1762–1779.

[12]Lauderdale, T. A., Bosson, C. S., Chu, Y.-C., and Erzberger, H., "Integrated Autonomous Terminal and Enroute Operations with Wind Errors," *18th AIAA Aviation Technology, Integration, and Operations Conference*, 2018.

[13]Nikoleris, T., Erzberger, H., Paielli, R. A., and Chu, Y.-C., "Autonomous System for Air Traffic Control in Terminal Airspace," *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014.

[14]Nikoleris, T., Erzberger, H., Paielli, R. A., and Chu, Y.-C., "Performance of an Automated System for Control of Traffic in Terminal Airspace," *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016.

[15]Cook, S. P. and Brooks, D., "A Quantitative Metric to Enable Unmanned Aircraft Systems to Remain Well Clear," *Air Traffic Control Quarterly*, Vol. 23, No. 2-3, 2015, pp. 137–156.

[16]Wu, M. G., Cone, A., Lee, S., and Edwards, M. W. M., "Well Clear Trade Study for Unmanned Aircraft System Detect and Avoid with Non-Cooperative Aircraft," *18th AIAA Aviation Technology, Integration, and Operations Conference*, 2018.

[17]Federal Aviation Administration, "Near Midair Collision Reporting," 2017, `http://www.faraim.org/aim/aim-4-03-14-530.html` (accessed April 16, 2018).

[18]Federal Aviation Administration, "Wake Turbulence Recategorization," 2016, `https://www.faa.gov/documentLibrary/media/Order/JO_7110_123_(2).pdf` (accessed April 16, 2018).