Combining Strategic and Tactical Conflict Management in Urban Air Mobility Environments

Laura Humphrey
NASA's Langley Research Center
Hampton, VA, USA
laura.r.humphrey@nasa.gov

César Muñoz
NASA's Langley Research Center
Hampton, VA, USA
cesar.a.munoz@nasa.gov

María Consiglio
NASA's Langley Research Center
Hampton, VA, USA

George Hagen
NASA's Langley Research Center
Hampton, VA, USA
george.hagen@nasa.gov

James Chamberlain
Analytical Mechanics Associates, Inc.
Hampton, VA, USA

Marco A. Feliú
Analytical Mechanics Associates, Inc.
Hampton, VA, USA
marco.feliu@nasa.gov

Richard Jessop
Analytical Mechanics Associates, Inc.
Hampton, VA, USA

***Abstract*—As Urban Air Mobility (UAM) evolves toward high-density operations, ensuring safety while minimizing the impact to air traffic control becomes critical. Toward that end, this paper explores a combination of strategic and tactical conflict management approaches to help maintain safe separation between aircraft in a congested UAM environment through fast-time simulations. Simulations are performed in Class B airspace near New York City using a proposed set of vertiports and routes at tempo and density levels expected in midterm UAM operations. Two pre-flight strategic conflict management (SCM) methods are compared: a baseline scheduler that considers only vertiports and an enhanced scheduler that also considers merging and crossing intersections. For tactical conflict management (TCM), an ownship-centric detect and avoid (DAA) capability that is parameterized by a well-clear volume is developed and compared over two different volumes. Results from these simulations help answer questions in UAM environments about the effectiveness of different SCM methods, how much additional improvement a TCM capability provides, what the tradeoffs are in terms of nuisance alerts versus loss of separation for more or less conservative TCM parameters, where conflicts tend to occur, and how often secondary conflicts arise after the TCM capability alters the course of an aircraft to avoid an initial conflict.**

***Index Terms*—Conflict Management, Tactical Conflict Management, Detect and Avoid (DAA), Urban Air Mobility (UAM)**

I. Introduction

Advanced Air Mobility (AAM) is a concept aimed at developing an air transportation system that moves people and cargo between local, regional, intraregional, and urban locations that are often underserved by aviation, using innovative aircraft, technologies, infrastructure, and operations. Urban Air Mobility (UAM) is a subset of AAM that focuses on flight operations in and around urban areas. UAM operations are anticipated to be high-tempo and high-density, leading to concerns that current air traffic services (ATS) will be insufficient to safely meet demand.

Future concepts therefore envision that higher frequency UAM operations will be enabled in part by defined Cooperative Areas (CAs), e.g., UAM corridors. These CAs will leverage collaborative techniques and technologies and highly automated cooperative flow management approaches that allow aircraft, eventually including remotely piloted and autonomous aircraft, to safely operate at increased operational tempos [1]. These envisioned technologies, techniques, and approaches include both strategic conflict management (SCM) services, which proactively attempt to prevent conflicts between aircraft trajectories by planning appropriate flight plans before take-off, and tactical conflict management (TCM) services, which attempt to resolve unexpected conflicts reactively during flight.

In a previous collaboration with the FAA William J. Hughes Tech Center, a series of simulations of UAM airport transfer operations were conducted in the Class B airspace near New York City at increased density and tempo levels representative of what is expected in midterm UAM operations [2]. The simulated scenarios included a carefully designed route structure for airport transfers, along with an SCM method to reduce the need for tactical separation in nominal operations. Preliminary observations of these simulations showed that despite the use of strategic deconfliction, unexpected conflicts can still arise due to operational and environmental uncertainty.

Motivated by these initial findings, this paper explores a combination of SCM and TCM for maintaining safe separation of aircraft through fast-time simulations in a congested UAM environment at tempos and traffic densities anticipated in midterm operations. Simulations are performed using two different SCM methods. One is a *baseline* capability that

This work was funded by NASA's Air Mobility Pathfinders (AMP) project, and funding for DASC participation was provided by NASA's Air Traffic Management & Safety (ATMS) project. The third, fifth, and seventh author's affiliations are shown at the time of their contributions to this work.

only considers scheduling flights so that they are deconflicted at vertiports, and the other is an *enhanced* capability that schedules flights so that they are also deconflicted around merging and crossing route intersections. Additional simulations are performed using the two different SCM methods in combination with a parameterized TCM capability instantiated with two different sets of separation parameters. One set is based on current detect and avoid (DAA) performance standards, and the other is chosen to be less conservative in order to reduce the number of unnecessary alerts in the more congested UAM environment. These simulations were designed to study questions such as:

- In terms of maintaining safe separation, what does the addition of the TCM capability over SCM provide?
- What are the tradeoffs between more and less conservative parameters for the TCM capability?
- Are there specific locations or situations in which tactical conflicts tend to occur?
- How often do secondary conflicts arise when the TCM capability alters the course of an aircraft to avoid an initial conflict?

In what follows, Section II defines some terms and concepts used throughout the work. Section III describes the two SCM methods, and Section IV describes the TCM capability. Section V details the simulation study, Section VI presents and discusses the results, and Section VII concludes the paper.

II. Definitions

This section briefly describes terms and concepts relevant to the conflict management approaches used in this paper and the results of the simulation study.

The term *well clear* as in 14 CFR 91.111 and 91.113 [3] refers to the legal requirement for an aircraft to maintain a safe separation distance from other aircraft to avoid collision risks. While there is no precise regulatory definition of well clear, there have been efforts by standards organizations such as RCTA Special Committee 228 to mathematically define this term and other related concepts in order to support the development of DAA capabilities for unmanned air systems (UAS) [4]. The resulting term *DAA well clear* defines a boundary or *well-clear volume* (WCV) around an aircraft that is parameterized by spatial and temporal thresholds with horizontal and vertical components. If one aircraft ever occupies the WCV of another, it is considered to be a *loss of DAA well clear* (LoWC). Since collision risk varies depending on how far one aircraft encroaches into the WCV of another, a metric defining the *severity of loss of DAA well clear* (SLoWC) was also developed. This metric ranges from 0–100% and estimates the penetration into the WCV accounting for both horizontal and vertical separation, with 0% indicating well clear and 100% representing no horizontal and vertical separation, i.e., a collision. Table I shows a proposed mapping from SLoWC to FAA severity categories [5]. The term *near mid-air collision* (NMAC) refers to two aircraft being within 500 feet horizontally and 100 feet vertically of each other. For reference, the minimum SLoWC at which this can occur is about 70% for a 1500 foot horizontal by 450 foot vertical WCV and 37% for an 800 by 300 foot WCV (with the exact percentage depending on encounter geometry).

TABLE I: SLoWC (SC-228) FAA Severity Description

SLoWC (%)	Severity	Description
0–17	Minimal	Pilot attention is required, but no action is required
17–33	Minor	Pilot action is required, low risk analysis event
33–47	Major	Corrective action is required, medium risk analysis event
47–94	Hazardous	Near midair collision, high risk analysis event
94–100	Catastrophic	Midair collision

A *conflict* is a predicted future infringement of an aircraft's WCV based on the current positions and velocities of the involved aircraft. In this work, a *(well-clear) encounter* is a pair-wise conflict between two aircraft. A *conflict avoidance maneuver* is a maneuver designed to avoid the conflict. A *return to mission maneuver* is a maneuver designed to return to the nominal flight plan after a conflict. And an *autonomous maneuver*, depicted in Fig. 1, is used to describe a sequence of conflict avoidance and return to mission maneuvers.

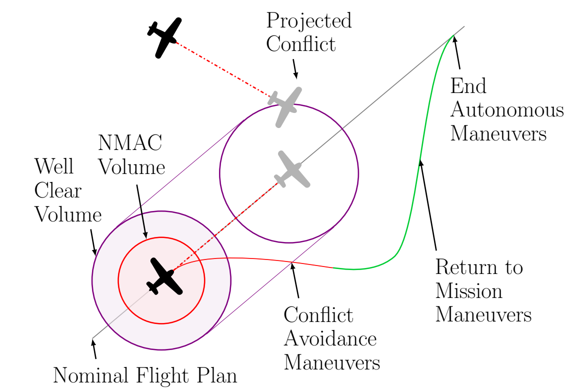


Fig. 1: Autonomous maneuvers to avoid a conflict.

III. Strategic Conflict Management

Two SCM methods were developed on top of a mission planning service, provided by NASA's ATS-TIGAR [6], to ensure that certain portions of every new flight plan are deconflicted with portions of existing flight plans before departure of the new flight. One is a baseline method that deconflicts flight plans at vertiports, and the other is an enhanced method that additionally deconflicts flight plans at points where routes in the airspace merge or cross.

Both SCM methods and the mission planning service as used in this work make the following assumptions. There is a given set of vertiports connected by pairs of uni-directional routes, one for each direction. Between routes, there is enough lateral separation to accommodate a safe separation distance between aircraft flying in opposite directions, plus a buffer to accommodate for flight technical error. Part of the airspace is designated as a CA within the Class B airspace, in which aircraft operators are primarily responsible for providing separation services. It is assumed that ATS provides safe separation services in the rest of the Class B airspace. Points at which routes cross the border of the CA are labeled as *CA entry* and *CA exit* points, and points at which routes cross or merge within the CA are labeled as *CA intersections*. In terms of vertiport infrastructure, each vertiport has one or more vertipads and optionally some number of parking spaces. Some vertiports are restricted to only one takeoff or landing operation at a time, either because there is only one pad or there are multiple pads with unfavorable geometry. This is referred to as a *single-in/single-out* (SISO) restriction. At the rest of the vertiports, simultaneous operations at different pads are allowed, either because pads are dedicated to only takeoffs or only landings, or the pads are far enough apart that simultaneous takeoffs and landings from different pads are feasible. For all vertiports and vertipads, it is assumed that a fixed amount of time is required for an aircraft to complete a landing operation, and a fixed amount of time is required for a takeoff operation. This includes enough time for the aircraft to either land at or takeoff from a vertipad and to optionally taxi on or off the pad if it has to move to or from a parking space or between takeoff- and landing-only pads.

It is further assumed that a customer makes a trip request from a specific operator, where each operator has its own independent fleet of aircraft, and individual aircraft can have different passenger capacities and performance characteristics.[1] The trip request specifies the origin and destination vertiports and number of passengers, and it is fulfilled on a first-come, first-served basis. When a request is received, the mission planner identifies a suitable aircraft for the trip, which might require a *repositioning flight* to move an aircraft to the origin vertiport if a suitable one is not already there. Given that vertiports have limited numbers of vertipads and parking spaces, if the origin vertiport is full, then a *clearing flight* is also required to move an unrelated aircraft to another vertiport to make room. A clearing flight might similarly be required for the destination vertiport. Some vertiports may be *staging vertiports* dedicated to support these "non-revenue" fleet movements, i.e., movements that do not serve passengers.

For each flight, the mission planner generates a flight plan as a 4D trajectory encoded in the Efficient Universal Trajectory Language (EUTL) [7]. Each flight plan is a kinematically feasible, waypoint-based trajectory that accounts for basic performance characteristics of the aircraft such as maximum vertical and longitudinal acceleration, maximum bank angle, and nominal climb, descent, and cruise speeds. It also accounts for desired angles of departure and arrival at each vertiport and follows predefined routes between vertiports.

The baseline SCM method uses a relatively simple approach to deconflict flight plans at vertiports. Given the times required for takeoff and landing operations, it models time windows of those durations around the corresponding operations of every flight plan. For vertiports without a SISO restriction, time windows for all operations across flight plans at a given vertipad are checked for overlap. For vertiports with a SISO restriction, the check is performed for all operations across flight plans at the vertiport. When a new flight plan is generated, if there are any overlaps with existing flight plans, the baseline SCM delays its takeoff time until all overlaps are resolved.

The enhanced SCM method additionally deconflicts flight plans at CA crossing and merging points. There is a precalculated nominal time window to clear each such point based on the relevant worst-case WCV and route geometry. For each flight plan, the enhanced SCM method calculates an ETA and clearing window as the aircraft moves over each point. When a new flight plan is generated, it checks to see whether the clearing windows of the corresponding aircraft would cover any crossing or merging points at the same time as those of any other aircraft. If so, it delays the takeoff time of the new flight plan until all overlaps are resolved. This is a minor variation of the resource scheduling approach described in [8].

IV. Tactical Conflict Management

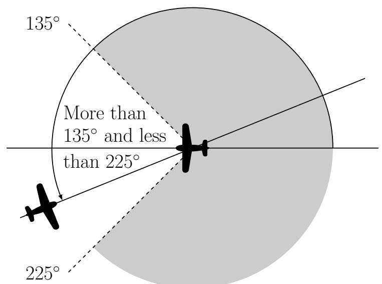
While the SCM approaches in Section III deconflict flight plans at vertiports (baseline) and crossing/merging points (enhanced), conflicts can still occur during operation. Conflicts can arise when aircraft significantly deviate from their flight plans due to unexpected events such as weather avoidance maneuvers, but they can also arise from relatively minor, expected deviations due to factors such as navigation precision and small variations in scheduled departure. TCM is therefore used to handle conflicts during flight.

The TCM capability developed for this work leverages NASA's DAIDALUS [9] (Detect and Avoid Alerting Logic for Unmanned Systems), a library of formally verified algorithms that serves as the reference implementation for RTCA's DAA Minimum Operational Performance Standards for UAS (DO-365) [10]. DAIDALUS, which is available as open-source software[2], has functions to determine the current well clear status of pairs of aircraft, check for conflicts within a given alerting time, and provide maneuver guidance to avoid LoWC or regain well clear status if it has been lost. To determine well clear status of the *ownship* on which it runs relative to an *intruder* aircraft, DAIDALUS uses a WCV parameterized by lateral and vertical distances along with a time-based threshold in the direction of relative horizontal velocity of the two aircraft. When the time threshold is zero (as in this work), the WCV is a cylinder. A non-zero time threshold elongates the cylinder in the direction of the relative velocity, adding extra protection for high closure rate encounters. To check for conflicts, it projects the WCV forward for a specified alerting time assuming that both aircraft maintain their current velocity vectors, and it issues an alert if there would be any well-clear violations over the projection. DAIDALUS additionally

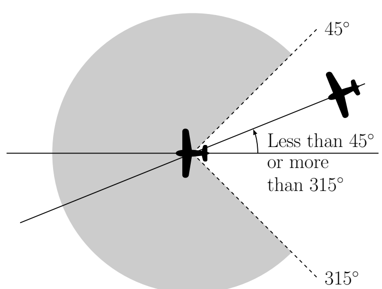
[1] In this work, there are two operators, each using a different type of aircraft. Notably, one has a 17 knot faster nominal cruise speed than the other.

[2] https://github.com/nasa/daidalus

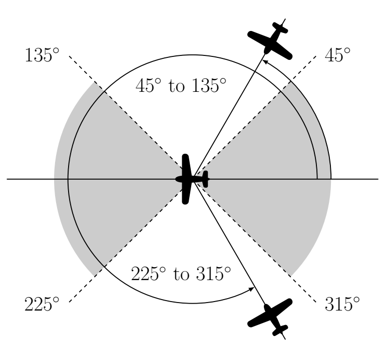
computes whether conflicts would (or would not) exist if the lateral direction (heading), vertical direction (vertical speed), altitude, or speed (airspeed) of the ownship's commanded velocity vector were changed to be within a range of new values, based on configurable values for aircraft horizontal/vertical acceleration, climb/descend rate, or turn rate/bank angle. By doing these calculations, DAIDALUS provides maneuver guidance in the form of ranges or *bands* of ownship headings, vertical speeds, altitudes, and airspeeds that will either cause, resolve, or avoid conflicts with intruders. Such guidance has been used in past work to provide assistive DAA guidance to pilots [11]. DAIDALUS has also been extended to include a sensor uncertainty mitigation logic to address sensor noise in position and velocity data [12].



(a) A head-on horizontal encounter.



(b) A same-track horizontal encounter.



(c) A crossing horizontal encounter.

Fig. 2: Three types of horizontal encounters.

In this work, the TCM capability runs DAIDALUS at fixed frequency time steps (1 Hz) to get updated detections, predictions, and guidance. At each time step, DAIDALUS determines whether there is a conflict with another aircraft within the specified lookahead time. If so, the TCM capability classifies the encounter in order to decide which aircraft has right of way. The encounter is classified as *vertical* if the aircraft are vertically separated by more than a vertical threshold (set to 300 feet) and as *horizontal* otherwise. Inspired by ICAO Document 4444 [13], horizontal encounters are further classified depending on the angular distance between tracks as *head-on* (135° to 225°), *crossing* (less than 45° or more than 315°), or *same-track* (all other angles), as depicted in Fig. 2. The TCM capability then uses rules inspired by 14 CFR 91.113 to determine the aircraft that has right of way. In a vertical encounter, it is the lower aircraft; in a head-on encounter, the slower aircraft; in a crossing encounter, the aircraft to the right; and in a same-track encounter, the leading aircraft.

If the ownship has right of way, the TCM capability has it continue at its current velocity vector for a configurable amount of time (set to 3 seconds). If there is still a conflict after this time has elapsed, it behaves as if the aircraft no longer has right of way. If the ownship does not have right of way, the TCM capability commands the aircraft to perform a conflict avoidance maneuver chosen from an appropriate DAIDALUS guidance band. If a loss of separation has occurred, the preferred maneuver is the one that guarantees the greatest separation at the minimum time. Otherwise, the preferred maneuver depends on the type of encounter. In a vertical encounter, the order of preference is vertical speed, heading, and airspeed. In a head-on encounter, the order of preference is heading, then vertical speed; airspeed is never used in a head-on encounter. In a crossing encounter, the order of preference is heading, airspeed, and vertical speed. In a same-track encounter, the order of preference is airspeed, vertical speed, and heading. DAIDALUS also provides altitude resolutions, but they are disabled for this study. For each modality of resolution, DAIDALUS computes a preferred resolution: the closest to the current ownship state. Fig. 3 illustrates a horizontal head-on encounter with a preferred resolution of 140° (the current heading of the aircraft is 126°).

The case of simultaneous conflicts with multiple aircraft is rare even for the high-density scenarios considered in this study. If this occurs, the logic classifies the encounter according to the most urgent one, i.e., the one that is closest in time, and resolves it accordingly. It should be noted that resolutions computed by DAIDALUS consider all possible conflicts with all traffic aircraft. So, any resolution, if it exists, solves all conflicts. Furthermore, DAIDALUS gives up computing a resolution if configurable minimum horizontal and vertical distances cannot be guaranteed (set to 300 feet horizontally and 100 feet vertically).

During conflict avoidance maneuvers, the TCM capability continues to monitor progress of the aircraft along its flight plan, recording the current next waypoint as *reached* if the aircraft comes within a certain distance of it or crosses the plane perpendicular to the flight plan through the waypoint. It

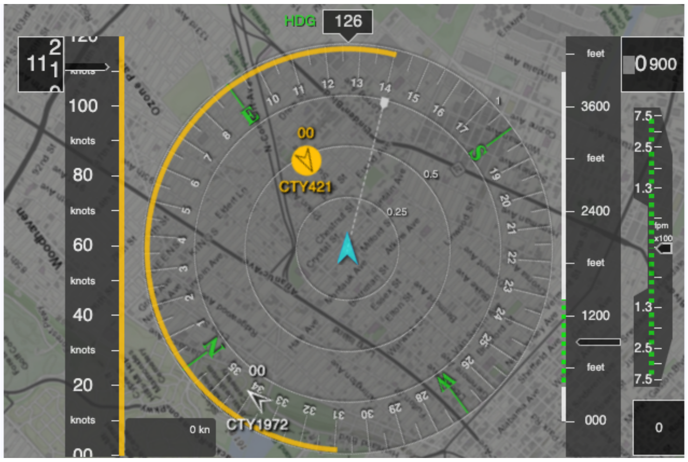


Fig. 3: Horizontal head-on encounter with preferred resolution.

checks at each time step whether changing heading toward the next waypoint would cross over any unsafe heading bands. If it would, then it continues to command the aircraft to follow its current velocity vector if it has no conflicts and to follow a conflict avoidance maneuver otherwise. If it would not, then the conflict avoidance maneuver ends, and the TCM capability starts return to mission maneuvers. During return to mission maneuvers, at each time step it commands the aircraft to follow a heading that points toward the next waypoint at the current ground speed and at a vertical speed equal to the difference between the aircraft's current altitude and that of the next waypoint times a small gain (set to 0.1), i.e., proportional control over altitude error. It also continues to update the next waypoint as waypoints are reached. If a waypoint is not reached within a given *persistence time* (set to 10 s), the next waypoint is incremented to avoid a lack of progress. The last *valid* waypoint is set to the last waypoint in the flight plan that has a lateral distance greater than 164 feet (50 meters) to the landing site. If the aircraft has not rejoined the flight plan by that point, i.e., reduced its cross-track error to below a threshold, then it makes an *emergency landing*. In the simulation framework used here, an emergency landing is currently coded to simply log the event and remove the simulated aircraft from the scenario. However, no emergency landings occurred in this study.

V. Simulation Study

In order to study the effects of different configurations and combinations of the SCM and TCM capabilities, a series of fast-time simulations were conducted. The simulations consist of UAM airport transfer operations with electric vertical takeoff and landing (eVTOL) vehicles in the Class B airspace near New York City using route structures that were designed as part of the prior simulation study [2].

A. Airspace Structure

The simulated airspace, shown in Fig. 4, includes nine vertiports, four of which are existing infrastructure and five of which are proposed. Table II shows the location, name, ID, number of vertipads, and number of parking spaces for each vertiport. Several vertiports have restrictions or features discussed in Section III. Vertiports KJRA, 6N5, and VJAM are subject to a SISO restriction; VFBF and VWSD each have only two vertipads, one for takeoffs and the other for landings; and VHIL, VFBF, and VWSD are staging vertiports used only for clearing and repositioning flights. A section of the airspace is designed as a CA in which cooperative Extensible Traffic Management (xTM) services are used to help maintain safe separation between aircraft. White lines indicate unidirectional tracks within the CA that are laterally separated by approximately 2000 feet and flown at a nominal cruise altitude of 900 feet MSL. This spacing was partly motivated by the current DO-365 WCV for terminal areas, which specifies a horizontal separation of 1500 feet. The routes incorporate existing defined helicopter routes, augmented by new routing informed by observed traffic patterns for flights traversing from the Hudson River to the south of Long Island. There are three entry/exit points where routes transition between the CA and traditional ATS environment, and within the CA, there are four sets of intersections between routes.

TABLE II: Vertiport Details

Location	Vertiport	ID	Vertipads	Parking Spaces
Downtown	East River	6N5	4	0
	Hudson River	KJRA	6	0
	Wall Street	KJRB	1	13
	Woodside	VWSD	2	20
Off Airport (CA/xTM)	Jamaica Station	VJAM	4	0
	Floyd Bennet Field	VFBF	2	20
	Hillside	VHIL	10	0
On Airport (ATS)	Rampside	VJKR	1	4
	Streetside	VJKS	2	0

B. Simulation Framework

Simulations were run using NASA's Simulation Infrastructure for Research on Interoperating Unmanned Systems (SIRIUS) [14], a flexible, highly configurable framework for simulation and analysis of future UAM operation concepts. In this study, SIRIUS takes as inputs for each aircraft a scheduled flight plan as computed by the SCM capabilities described in Section III, an aircraft dynamics model, and a decision-making and control module that implements a basic guidance and control law for following flight plans and the TCM capability described in Section IV for detecting conflicts and performing autonomous maneuvers in response. SIRIUS advances the simulations in discrete time steps of configurable length (set to 0.1 s). At each time step, SIRIUS passes the state of other simulated aircraft to each aircraft's decision-making and control module, which computes new heading, vertical speed, and ground speed commands for the aircraft. These commands are passed to the aircraft dynamics model, which is used to update the aircraft's state in response. The process repeats until all aircraft have landed. SIRIUS includes

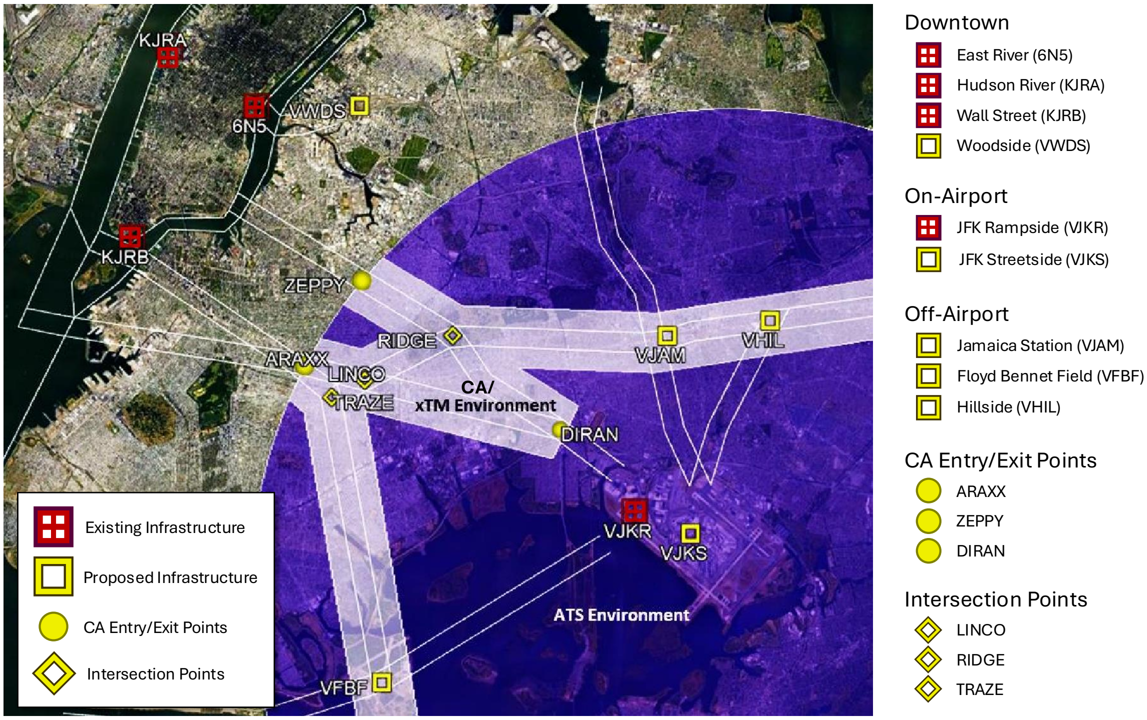


Fig. 4: New York City Class B airspace used in the simulation study.

capabilities for modeling wind, communication delays and dropouts, and sensor uncertainties, but these capabilities are not used in this study. SIRIUS computes several DAA metrics over each entire flight, but because this study is focused only on the portions of flight in the cooperative area, metrics are computed using ATS-TIGAR post-processing scripts from log information provided by SIRIUS.

The aircraft model used in this study is based on a six-passenger quadcopter eVTOL model designed to study the performance of various eVTOL design concepts [15]. The model computes the power required for a heading, airspeed, and vertical speed command given the aircraft's current state and number of passengers. If the power required exceeds power available, it includes logic to throttle commanded airspeed and vertical speed in proportion to the requested acceleration for each. However, in these simulations with four passengers per aircraft, the command was very rarely throttled. This was mostly for brief periods of time during the climb phase of the flight plan, and it had little observable effect on the trajectory compared to running the model with no power limit. Given a command, changes in the aircraft's airspeed v, vertical speed $\dot{h}$, heading ψ clockwise from north, and bank angle ϕ are modeled as a simple proportional controller

$$\dot{v} = k_v(v_{cmd} - v) \tag{1}$$

$$\ddot{h} = k_h(\dot{h}_{cmd} - \dot{h}) \tag{2}$$

$$\Delta\psi_{des} = k_\psi(\psi_{cmd} - \psi) \tag{3}$$

$$\phi_{des} = \tan_d^{-1}(v\ \Delta\psi_{des}/\mathrm{g}) \tag{4}$$

$$\dot{\phi} = k_\phi(\phi_{des} - \phi) \tag{5}$$

$$\dot{\psi} = \tan_d(\phi)\ \mathrm{g}/v \tag{6}$$

where g is the gravitational constant, tangent $tan_d(\cdot)$ and inverse tangent $\tan_d^{-1}(\cdot)$ operate over degrees, and gains are set to $k_h = k_v = k_\phi = 1.0\ \mathrm{s}^{-1}$ and $k_\psi = 0.3\ \mathrm{s}^{-1}$. There are also bounds on elements of the aircraft's state. For the variables, bounds are $|\dot{h}| \leq 2000$ feet per minute, $|\Delta\psi_{des}| \leq 8°$ per second, and $|\phi_{des}| \leq 30°$. For their derivatives, bounds are $|\ddot{h}| \leq 0.1\mathrm{g}$, $|\dot{v}| \leq 0.1\mathrm{g}$ or $0.12\mathrm{g}$ depending on the model's current flight segment mode, and $|\dot{\phi}| \leq 20°$ per second.

During nominal flight, discrete-time controllers for ground track, vertical speed, and ground speed were developed to command the eVTOL model to follow the aircraft's assigned flight plan. Each controller tracks progress of the aircraft along its flight plan in EUTL format in the same way as the TCM capability described in Section IV. Commands are then computed over the segment between the most recently reached waypoint and the next waypoint. Every waypoint in a EUTL flight plan represents a *trajectory change point*. Along with a position and time, each waypoint specifies the start or end of a given longitudinal or vertical acceleration or of a circular turn segment with a given direction, radius, and center. Also note that flight plans in this study can be partitioned into three sequences of waypoints: one for takeoff/climb, followed by one for level flight between vertiports, followed by one for landing/descent. All flight plans therefore have one waypoint marked as *top of climb* (TOC) and one as *top of descent* (TOD). Finally note that in these simulations, there is no wind, so ground track is equivalent to heading, and ground speed is equivalent to airspeed. Otherwise, the commands would need to be modified to account for wind.

The ground track controller is based on [16]. To rejoin a line segment pointed north, the desired aircraft track angle (all controller ground track angles are in radians clockwise from north) as a function of cross-track error y (meters positive to the right) and an underlying vector field shaped by parameters χ^∞ and K is

$$\chi^d(y) = -\chi^\infty \frac{2}{\pi} \tan^{-1}(Ky) \tag{7}$$

where χ^∞ is the angle of the vector field when cross-track error is large, and K is a positive constant that sets the vector field's rate of transition from χ^∞ to 0 as cross-track error goes to 0. The commanded track angle χ^c is then

$$\chi^c = \chi - \frac{1}{\alpha}\chi^\infty \frac{2}{\pi} \frac{K}{1+(Ky)^2} v \sin(\chi) - \frac{\kappa}{\alpha} \operatorname{sat}\left(\frac{\tilde{\chi}}{\epsilon}\right) \tag{8}$$

where χ and v are current aircraft track angle and ground speed, $\tilde{\chi} = \chi - \chi^d(y)$ in $[-\pi, \pi)$, $\operatorname{sat}(\cdot)$ saturates its argument at ± 1, and α and ϵ are gains for an additional sliding mode control term. χ^c is then adjusted for the actual segment angle. For counterclockwise turn segments, the desired track angle is

$$\chi^d(d) = \gamma -_{ccw} \frac{\pi}{2} -_{ccw} \tan^{-1}(K(d-r)) \tag{9}$$

where r is the turn's radius, d and γ are distance and angle of the aircraft from the turn's center, and $-_{ccw}$ changes from subtraction to addition for clockwise turns. Commanded track angle is then

$$\chi^c = \chi + \frac{v}{\alpha d} \sin(\chi - \gamma) -_{ccw} \frac{\beta}{\alpha} v \cos(\chi - \gamma) - \frac{\kappa}{\alpha} \operatorname{sat}\left(\frac{\tilde{\chi}}{\epsilon}\right) \tag{10}$$

where $\beta = K/(1+(K(d-r))^2)$. For this aircraft model, controller parameters and gains in (7)–(10) were set to $\chi^\infty = \pi/2$ rad (as in [16]), $K = 0.003$ m^{-1}, $\alpha = 0.4$ rad/s, $\kappa = 0.6$ rad^2/s, and $\epsilon = 1.5$ rad. Gain ϵ, which modulates sliding mode terms near the flight plan, was found to produce a steady oscillatory response (sometimes 180° in the wrong direction) when its value is too small. Simulations were therefore performed over the level flight portion of several different flight plans with the aircraft initialized to have a random position and heading error to find a value of ϵ that did not produce oscillations. The rest of the gains and parameter K were selected by manually searching for reasonable initial values, then running parameter sweeps around those values over a set of nine representative flight plans. The values used in this study are those that produced relatively small maximum and cumulative cross-track error while limiting heading oscillations along straight line segments.

The vertical speed controller is a simple proportional integral (PI) feedback controller over vertical position error plus a feedforward term for speed, with a limit of 750 feet per minute. Let h_0 and $\dot{h}_0$ be the planned vertical position and speed at the start of the segment, $\ddot{h}_{0,1}$ the planned constant vertical acceleration (potentially 0) over the segment, t_0 and t_1 the planned start and end times for the segment, k_0 the simulation time step at which the aircraft reached the start of the segment, k the current time step, and $t = (k - k_0)\Delta t_s$ the amount of time the aircraft has been traveling the segment. Based on the general equation for speed as a function of initial speed, acceleration, and time

$$v(t) = v_0 + at, \tag{11}$$

the feedforward term for vertical speed is

$$\dot{h}_{ff}[k] = \dot{h}_0 + \ddot{h}_{0,1} \cdot \min((k-k_0)\Delta t_s, t_1 - t_0). \tag{12}$$

The desired vertical position for the feedback term is computed differently depending on the vertical slope of the segment. For segments where vertical slope is high (the horizontal component is less than 1 meter long or the ratio of vertical to horizontal components is greater than 3.5), it is computed according to the general equation for position as a function of initial position, velocity, acceleration, and time

$$p(t) = p_0 + v_0 t + at^2, \tag{13}$$

so that

$$\begin{aligned} h_{des}[k] = \; & h_0 + \dot{h}_0 \cdot \min((k-k_0)\Delta t_s,\, t_1 - t_0) \\ & + \ddot{h}_{0,1} \cdot \min((k-k_0)\Delta t_s,\, t_1 - t_0)^2. \end{aligned} \tag{14}$$

For segments with low vertical slope, $h_{des}[k]$ instead corresponds to the vertical position at the aircraft's projection onto the flight plan in the horizontal plane, to keep the aircraft's vertical and horizontal positions relative to the flight plan synchronized. In both cases, the vertical feedback term is

$$e_h[k] = (h_{des}[k] - h[k]) \tag{15}$$

$$\dot{h}_{fb}[k] = K_{p_h} e_h[k] + K_{i_h} \, \Sigma_{j=j_0}^{k-1} e_h[j] \Delta t_s \tag{16}$$

where j_0 in the integral term is the most recent time step at which the controller was enabled or re-enabled, i.e., on takeoff and after control is returned by the TCM capability. As with ground track, gains $K_{p_h} = 0.5$ s^{-1} and $K_{i_h} = 0.008$ s^{-2} in (16) were selected by manually searching for reasonable initial values, then running parameter sweeps over the climb and initial level flight phases of a representative flight plan. Final gain values were selected to minimize cumulative error while keeping overshoot low over the initial level flight phase.

The ground speed controller has feedback and feedforward terms similar to those of the vertical speed controller, except that it has an extra mode for the feedback term that adjusts speed to help the aircraft reach waypoints at the planned time if it gets off schedule. The commanded speed is limited to a minimum v_{min} of 0.1 knots and v_{max} of 120 knots. The feedforward term is always

$$\dot{v}_{ff}[k] = \dot{v}_0 + \ddot{v}_{0,1} \cdot \min((k-k_0)\Delta t_s, t_1 - t_0) \tag{17}$$

where $\dot{v}_0$ is the planned ground speed at the start of the segment, $\ddot{v}_{0,1}$ the planned constant ground acceleration (potentially 0) over the segment, and the other terms are as in the vertical speed controller. The feedback term depends on the flight plan segment. For non-vertical segments with non-zero longitudinal acceleration, it is a PI feedback control term over the error in actual versus planned distance traveled. Planned distance traveled along the segment $d_{des}[k]$ can be computed using (13), and actual distance traveled $d[k]$ can be computed by projecting the aircraft's position onto the closest point on the segment. The feedback term is then

$$e_d[k] = (d_{des}[k] - d[k]) \tag{18}$$

$$v_{fb}[k] = K_{p_v} e_d[k] + K_{i_v} \Sigma_{j=j_0}^{k-1} e_d[j] \Delta t_s. \tag{19}$$

Time step j_0 in the integral term is reset whenever the previous segment was a vertical segment, a segment in which there was no longitudinal acceleration, or control was returned from the TCM capability. For non-vertical segments with zero longitudinal acceleration, a different feedback term is used to adjust for schedule. The time remaining $t_{remain}[k]$ to the next waypoint is computed based on the planned speed and distance remaining $d_{remain}[k]$, along with how far behind schedule $t_{behind}[k]$ the aircraft would reach the next waypoint at the planned speed (negative when the aircraft is ahead of schedule). If $t_{remain}[k] > t_{buffer}$ and $t_{remain}[k] > t_{behind}[k]$, then

$$v_{fb}[k] = v_{max} - v_{ff}[k] \tag{20}$$

so that $v[k] = v_{ff}[k] + v_{fb}[k] = v_{max}$ to catch up as best as possible. Otherwise, if $t_{remain}[k] > t_{buffer}$ and $t_{delay}[k] < t_{remain}[k] + \epsilon$, then

$$v_{fb}[k] = d_{remain}[k]/(t_{remain}[k] - t_{delay}[k]) - v_{ff}[k] \tag{21}$$

to reach the waypoint on time. In all other cases, $v_{fb}[k] = 0$. The effect of t_{buffer} (set to 5 s) is to allow the feedback term to drop to zero when close to the next waypoint so that the aircraft returns to the planned speed before the next segment begins. Feedback is capped to a maximum value (set to 2.5 m/s) so that DAA assumptions on maximum aircraft speed are not violated. Similar to the vertical speed controller, the gains $K_{p_v} = 0.5$ s^{-1} and $K_{i_v} = 0.012$ s^{-2} in (19) were selected by simulating the controller over the landing phase of a representative flight plan and choosing values that minimized cumulative position error but had low ground speed near the landing site.

The decision-making and control module is implemented using NASA's Independent Configurable Architecture for Reliable Operation of Unmanned Systems (ICAROUS) [17]. ICAROUS provides several implementations of algorithms to support robust autonomous UAS operations, along with a decision-making framework that allows a user to define behaviors along with monitors and triggers that determine which behavior to execute. In this simulation study, the key monitors and triggers and their mapping to behaviors are:

- Flight start → Fly the nominal flight plan.
- The aircraft reaches the flight plan's top of climb → Enable TCM.
- The TCM detects a conflict → The TCM performs conflict avoidance maneuvers.
- The TCM detects that a conflict is over → The TCM performs return to mission maneuvers.
- The aircraft reaches the flight plan's top of descent while there are no conflicts → Disable TCM.
- During a conflict, the aircraft reaches the closest waypoint in the nominal flight plan outside a 500 m horizontal ring around the landing site → Make an emergency landing.
- No other conditions apply → Fly the nominal flight plan.

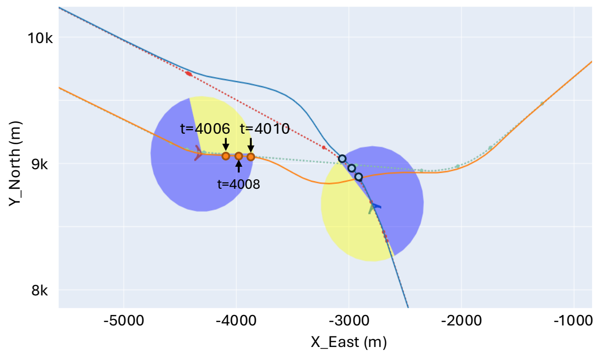
ICAROUS actions are completely autonomous and do not follow a pilot model, e.g., reaction time only depends on the event model architecture implemented by the system, and resolution maneuvers follow the hierarchy described in Section IV. In addition to the DAA and decision-making logics used in this study, ICAROUS supports keep-in/keep-out geofences and merging & spacing capabilities. It can also be used in a non-simulated mode and has been flown in several flight test campaigns [18]–[20].

VI. Simulation Results

A variety of metrics were computed for the scenario in Section V to compare performance of the two SCM methods with and without the TCM capability, both using two different WCVs. The simulations included 128 random trip requests over a two-hour time frame, with 64 in the first hour and 64 in the second. In these results, the baseline SCM method is referred to as SCM1 and the enhanced SCM method as SCM2. WCV1 refers to the 1500 by 450 foot WCV, and WCV2 refers to the 800 by 300 foot WCV. To give a sense of computational complexity of the SCM and TCM, for each condition, it takes just under a minute for the SCM to generate deconflicted flight plans for all requested, repositioning, and clearing flights and to perform some preliminary analysis. Simulating the scenario for each condition in SIRIUS with the eVTOL model and TCM active takes 14–15.5 minutes with data logging and preliminary analysis enabled and 6–7 minutes with no data logging or analysis. The following metrics were then computed over a 60 minute time window starting 30 minutes into the simulation and only within the CA.

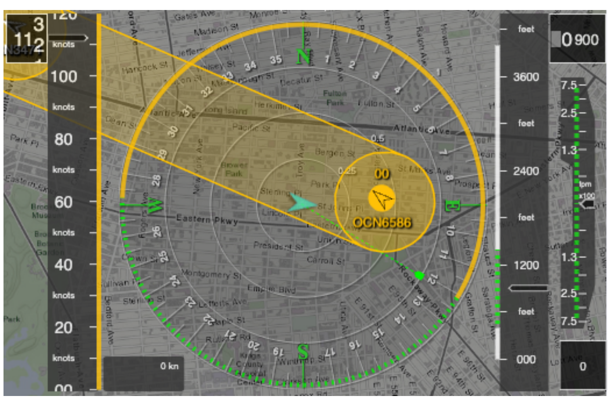
To give a sense of scale, Table III shows the average, maximum, and total number of flights active in the CA during the analysis window. Table IV shows the number of LoWC events by severity for each of the conditions, along with the minimum distance at the closest point of approach (DCPA) across all flights under that condition. These results show that SCM2 has fewer LoWC events than SCM1, and the use of TCM both reduces the total number and severity of LoWC events and increases minimum DCPA. Indeed, under TCM all losses are classified as minimal. The number of LoWC events is counted per aircraft. Therefore, it is usually even, since LoWC events tend to involve two aircraft. In cases where the number is not even, it is either because more than two aircraft were involved, or as in this case, only one of two aircraft registered a LoWC because one is in the CA during the LoWC and the other is not.

An inspection of the simulation traces showed that for both TCM conditions in which there were LoWC events, there was



(a) Before a conflict leading to a LoWC at t=4002 s.

(b) Conflict and avoidance maneuver at t=4006 s.



(c) Conflict and avoidance maneuver at t=4010 s.

Fig. 5: A typical minimal LoWC event with TCM enabled.

a common cause for all but one instance. The LoWC events were all at an intersection where the aircraft that did not have right of way was commanded to perform collision avoidance maneuvers based on heading, but the commanded turn was in the direction opposite to a turn the aircraft had made moments before. Recall that the aircraft model as described in Section V models turn rate as a function of bank angle and also limits bank angle rate. The aircraft is therefore still banking in the opposite direction when the conflict starts and cannot change heading as quickly as the configured vehicle turn rate used by the DAA logic. Fig. 5a shows how one such event unfolds. Fig. 5b shows the status at the start of the conflict at $t = 4006$ s. The encounter is initially classified as a crossing encounter but is re-classified as a head-on encounter at $t = 4008$ s. In both cases, OCN6586 has right of way, so its TCM capability commands it to turn right as it is coming out of a left turn. The dashed green bands along the right are recovery bands for vertical speed and altitude maneuvers, meaning that there is not enough time to avoid a loss using those maneuvers. Unfortunately, as Fig. 5c shows, the response of the aircraft lags behind the heading command, so that heading recovery bands are generated at $t = 4010$ s, and a LoWC can no longer be avoided, even though OCN6586 starts a conflict avoidance maneuver just prior to this point as well. The one LoWC that did not follow this same pattern was one of the conflicts for the SCM2+WCV2 with TCM condition. It also occurred at an intersection, but a change of classification from head-on to crossing during the encounter changed which aircraft had right of way, delaying the response.

TABLE III: Number of Active Flights

Simulation Condition	Avg, Max, Total Flights Active (no TCM)	Avg, Max, Total Flights Active
SCM1+WCV1	2.7, 7, 74	2.8, 7, 75
SCM1+WCV2	2.5, 7, 68	2.7, 7, 70
SCM2+WCV1	2.7, 6, 74	2.7, 6, 75
SCM2+WCV2	2.6, 6, 70	2.6, 6, 71

TABLE IV: LoWC Results

Simulation Condition	SCM LoWC Total (Minimal, Minor, Major, Hazardous, Catastrophic)	SCM+TCM LoWC Total (Minimal, Minor, Major, Hazardous, Catastrophic)
SCM1+WCV1	16 (0, 0, 4, 6, 6) Min DCPA: 130 ft	6 (6, 0, 0, 0, 0) Min DCPA: 1370 ft
SCM1+WCV2	0 (0, 0, 0, 0, 0) Min DCPA: N/A	0 (0, 0, 0, 0, 0) Min DCPA: N/A
SCM2+WCV1	12 (6, 4, 0, 2, 0) Min DCPA: 130 ft	3 (3, 0, 0, 0, 0) Min DCPA: 706 ft
SCM2+WCV2	2 (0, 0, 0, 0, 2) Min DCPA: 5 ft	0 (0, 0, 0, 0, 0) Min DCPA: N/A

Table V shows the number of alerts for conflict detection events in each condition. A *nuisance alert* refers to alerts that do not have a corresponding LoWC when no conflict avoidance maneuver is made. Since the TCM always maneuvers in response to an alert, nuisance alerts do not apply to TCM conditions. These results show that for the SCM-only conditions, WCV2 produces fewer alerts than WCV1. This is unsurprising since WCV2 is smaller than WCV1. The results also show that in the SCM plus TCM conditions, the total number of alerts tends to go up, even more in the WCV1 than WCV2 conditions.

TABLE V: Alerts

Simulation Condition	SCM Nuisance Alerts / SCM Total Alerts	TCM Total Alerts
SCM1+WCV1	27/43	66
SCM1+WCV2	35/36	66
SCM2+WCV1	1/13	18
SCM2+WCV2	3/5	5

Table VI shows maximum cross-track and vertical deviation from any flight plan, and maximum delay in arrival to any flight plan's TOD. As expected, these values are higher with SCM1 compared to SCM2. Simultaneous conflicts were also counted but, interestingly, there were none with TCM.

TABLE VI: Maximum Deviations

Simulation Condition	**Max Cross-Track Deviation**	**Max Vertical Deviation**	**Max TOD Arrival Delay**
SCM1+WCV1	9007 ft	323 ft	481 s
SCM1+WCV2	8827 ft	99 ft	341 s
SCM2+WCV1	1377 ft	6 ft	118 s
SCM2+WCV2	79 ft	6 ft	9 s

VII. Conclusions

This paper reports preliminary results of a fast-time simulation study that explores a combination of SCM and TCM in a cooperative UAM airspace over routes between a set of proposed vertiports in New York City Class B airspace. For SCM, two different methods were employed, a baseline method that deconflicts flight plans at vertiports and an enhanced method that additionally deconflicts flight plans at intersections. For TCM, an ownship-centric DAA capability was employed. Two different WCVs were used, a 1500 by 450 foot volume as recommended by DO-365-B, and a less conservative 800 by 300 foot volume intended to reduce the number of nuisance alerts. In terms of the questions the study aimed to answer, the findings show that (1) the addition of TCM does in fact reduce the number of and severity of LoWC events, (2) a larger WCV results in higher numbers of alerts, higher ratios of nuisance alerts, and larger resolution maneuvers, but it also maintains a larger minimum DCPA between aircraft, (3) conflicts tend to occur near intersections, (4) secondary conflicts due to the TCM were not observed.

An additional finding is that when operating with the TCM capability, there were two causes for LoWC within the CA. The first is when the TCM commands a heading collision avoidance maneuver in the opposite direction of a turn that the aircraft is currently making or recently made. This is because the DAA logic assumes a simple kinematic model of the aircraft dynamics, where the ownship is capable of instantaneously changing its turn rate. The second is when encounter classification changes in the middle of an encounter, changing which aircraft has right of way and therefore delaying a consistent response. Also, though this work focuses on metrics within the CA, it was observed that for conflicts with TCM outside the CA, a frequent cause was that TCM on an aircraft had been disabled as it approached a vertiport, but the other aircraft was designated as the right of way aircraft, delaying a conflict avoidance maneuver. All these effects are artifacts of the simulation, which might be mitigated by enhancements to the DAA and right-of-way logics. Finally, though this work did not focus on the throughput of different SCM methods, it should be noted that prior work [6], [8] studied the effects on throughput and presented some other SCM strategies that might have reduced the number of conflicts. These findings suggest directions for possible future work.

References

[1] Federal Aviation Administration, "Urban air mobility (UAM) concept of operations 2.0," 2023.

[2] J. Prince, N. Guerreiro, C. Denham, T. McClain, R. Ghatas, and T. White, "Feasibility and scalability of midterm UAM operations in cooperative airspace: Class B airspace simulation evaluation," NASA, Tech. Rep. NASA/TM-20250009519, 2025.

[3] Federal Aviation Administration, "United States Code of Federal Regulations, Title 14 – Aeronautics and Space, Part 91 – General Operating and Flight Rules," 2025.

[4] A. C. Cone, D. P. Thipphavong, S. M. Lee, and C. Santiago, "UAS well clear recovery against non-cooperative intruders using vertical maneuvers," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017.

[5] V. A. Carreño, M. Consiglio, and C. A. Muñoz, "Analysis and preliminary results of a concept for detect and avoid in the cockpit," in *IEEE/AIAA 38th Digital Avionics Systems Conference*, 2019.

[6] N. M. Guerreiro, R. W. Butler, J. M. Maddalon, and G. E. Hagen, "Mission planner algorithm for urban air mobility–initial performance characterization," in *AIAA Aviation 2019 Forum*, 2019.

[7] G. E. Hagen, N. M. Guerreiro, J. M. Maddalon, and R. W. Butler, "An efficient universal trajectory language," NASA, Tech. Rep. NASA/TM-2017-219669, 2017.

[8] N. M. Guerreiro, G. E. Hagen, J. M. Maddalon, and R. W. Butler, "Analysis of strategic conflict management approaches as applied to simulated UAM operations," NASA, Tech. Rep. NASA/TM–20220017551, 2023.

[9] C. Muñoz, A. Narkawicz, G. Hagen, J. Upchurch, A. Dutle, M. Consiglio, and J. Chamberlain, "DAIDALUS: detect and avoid alerting logic for unmanned systems," in *IEEE/AIAA 34th Digital Avionics Systems Conference*, 2015.

[10] I. RTCA, "Minimum operational performance standards (MOPS) for detect and avoid (DAA) systems," Special Committee No. 228, Tech. Rep. Document DO-365C, September 2022.

[11] J. Chamberlain, M. Consiglio, C. Muñoz, and P. Masci, "Assistive detect and avoid technology in urban air mobility environments," in *AIAA DATC/IEEE 43rd Digital Avionics Systems Conference*, 2024.

[12] A. Narkawicz, C. Muñoz, and A. Dutle, "Sensor uncertainty mitigation and dynamic well clear volumes in DAIDALUS," in *IEEE/AIAA 37th Digital Avionics Systems Conference*, 2018.

[13] International Civil Aviation Organization, *Procedures for Air Navigation Services — Air Traffic Management (PANS-ATM)*, 16th ed., International Civil Aviation Organization, Montréal, Quebec, Canada, 2016.

[14] A. Peters, B. Duffy, S. Balachandran, M. Consiglio, and C. Muñoz, "SIRIUS: Simulation infrastructure for research on interoperating unmanned systems," in *IEEE/AIAA 40th Digital Avionics Systems Conference*, 2021.

[15] D. Hartman, C. L. Hartman, and J. V. Foster, "Performance modeling of urban air mobility vehicles to support air traffic management research," in *AIAA SCITECH 2023 Forum*, 2023.

[16] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[17] S. Balachandran, C. Muñoz, M. Consiglio, M. Feliú, and A. Patel, "Independent configurable architecture for reliable operation of unmanned systems with distributed on-board services," in *AIAA/IEEE 37th Digital Avionics Systems Conference*, 2018.

[18] B. Duffy, S. Balachandran, A. Peters, K. Smalling, M. Consiglio, L. Glaab, A. Moore, and C. Muñoz, "Onboard autonomous sense and avoid of non-conforming unmanned aerial systems," in *AIAA/IEEE 39th Digital Avionics Systems Conference*, 2020.

[19] A. J. Moore, M. Schubert, N. Rymer, S. Balachandran, M. Consiglio, C. Muñoz, J. Smith, D. Lewis, and P. Schneider, "UAV inspection of electrical transmission infrastructure with path conformance autonomy and lidar-based geofences NASA report on UTM reference mission flights at Southern Company flights November 2016," NASA, Tech. Rep. NASA/TM-2017-219673, 2017.

[20] M. Consiglio, B. Duffy, S. Balachandran, L. Glaab, and C. Muñoz, "Sense and avoid characterization of the Independent Configurable Architecture for Reliable Operations of Unmanned Systems," in *13th USA/Europe Air Traffic Management R&D Seminar*, 2019.