An Experimental System for Strategic Flight Path Management in Advanced Air Mobility

David A. Karr, PhD* and James L. Sturdy†
Analytical Mechanics Associates, Hampton, VA 23666, USA

Bryan A. Barrows‡ and Mark G. Ballin§
NASA Langley Research Center, Hampton, VA 23681, USA

In the concept envisioned for Urban Air Mobility (UAM) operations, fleets of electric vertical takeoff and landing (eVTOL) vehicles would operate between vertiports distributed within a densely populated area. These operations would be largely independent from the existing air traffic control system and would place the responsibility for flight planning and aircraft separation on fleet operators. The fourth major level on the UAM Maturity Level scale, UML-4, relies on "collaborative and responsible" automation to enable operations in non-visual conditions with medium traffic density (hundreds of aircraft in one metropolitan region) and medium complexity. This level of service places many requirements on automation systems to assist the operators of these aircraft. NASA has developed the Autonomous Operations Planner (AOP), a reference prototype Flight Path Management automation system, and has modified AOP to support research of anticipated UML-4 operations. AOP creates a four-dimensional flight plan conforming to the constraints of these operations, evaluates and modifies the flight plan during flight as conditions and constraints evolve, and coordinates the flight plan with other airspace users and with service providers. This version of AOP has been integrated into the Sikorsky Autonomy Research Aircraft and used in a flight test activity. In this paper we discuss anticipated characteristics of UAM operations, modifications that were made to AOP to adapt to that environment or to support the flight test, and observations of software and aircraft performance during the flight test. The aircraft achieved four-dimensional conformance with the flight plan and AOP provided adequate planning in almost all cases. We discuss improvements that could be made to AOP to address deficiencies that were observed.

I. Introduction

Urban Air Mobility (UAM) operations, a subset of the concept of Advanced Air Mobility (AAM), poses difficult questions to be answered by research. NASA has been an active participant in this research, performing multiple metropolitan-area UAM simulation studies culminating in a flight test in October 2023 to study an advanced concept of Flight Path Management (FPM).

NASA's Autonomous Operations Planner (AOP) served as a reference prototype FPM software system for this activity. This paper focuses on the significant technical challenges in developing this prototype FPM system for a UAM operational environment and integrating it into flight test systems to achieve the research objectives. Reference [1] describes the entire research activity, including the batch and human-in-the-loop (HITL) simulation studies and the flight test. Reference [2] describes the design and execution of the flight test.

The observations in this paper illustrate how adaptable AOP is to radically new operational environments such as UAM. We also present some preliminary observations on the flight test, focusing on questions relevant to AOP.

Section II of this paper describes the principles of FPM and in particular how they apply to UAM. Section III describes the equipment selected for the flight test of the prototype FPM system and gives a brief overview of the individual test points that were performed. Section IV describes the modifications that we made to AOP to adapt it to the UAM simulation environment used in the batch and HITL studies and to prepare it for the flight test. Section V presents observations we made during the flight test or while analyzing data collected from the flight test. The observations in

---

*Lead Principal Scientist, AIAA Senior Member

†Lead Principal Systems Engineer

‡Aerospace Research Engineer, AIAA Member

§Aerospace Research Engineer, AIAA Associate Fellow this section focus on the functions of AOP itself; see Ref. [1] for a higher-level analysis of the system performance observed during the flight test. Section VI discusses implications of the observations made in Section V. Section VII summarizes the findings of this paper.

II. Flight Path Management

Before we describe the flight test and other research activities in greater detail, the following summary of the principles of FPM and its application to UAM sets these activities in context.

A. Concept

Flight Path Management was first defined in Ref. [3] in an effort to identify the various functions that are performed in managing the airborne phase of flight, whether performed by a human operator, automation, or a combination of human operators and automation. FPM provides the function of strategic flight path planning in the presence of other users sharing the airspace and in the presence of restrictions and changing constraints.

In FPM, *strategic planning* is defined as flight path planning that achieves mission objectives and retains them when replanning is necessary. Strategic planning may occur before flight or during flight. We may also refer to in-flight strategic planning as *strategic replanning*. Strategic planning can be contrasted with tactical planning, which suspends mission objectives in order to achieve a more important objective such as maintaining safety.

Aircraft flights are typically subject to numerous constraints. Depending on the nature of the operations of the aircraft, these constraints can include (but are not limited to) a required time of arrival (RTA) or an altitude constraint at a waypoint, regions of special-use airspace that must be avoided, areas of severe weather, terrain, separation from other aircraft, aircraft endurance, and the economics of operations. FPM considers all constraints simultaneously by strategically replanning the remaining flight each time a constraint changes or new constraint is encountered.

There are several alternatives for the implementation of the FPM function. The work described in this paper explores one alternative: equip many individual aircraft with technology that enables this functional capability. This onboard automation assists the pilot of each aircraft in executing FPM functions. Each aircraft shares information relevant to other system participants and receives information relevant to itself regarding other aircraft. Data exchange and any necessary coordination is accomplished through airborne and ground-based data links.

B. Application to Urban Air Mobility

In the concept envisioned for UAM operations, fleets of electric vertical takeoff and landing (eVTOL) vehicles would operate between vertiports distributed within a densely populated area. To facilitate advancement toward this vision, NASA has developed the UAM Maturity Level (UML) scale, which anticipates various levels of service and the capabilities expected at each level [4]. The fourth level on the scale, UML-4, is defined as hundreds of simultaneous aircraft aloft over the served urban area in a variety of weather conditions, including instrument meteorological conditions (IMC) and visual meteorological conditions (VMC). UML-4 corresponds to an extremely high level of traffic density compared to current operations. Air traffic management functions, which ensure aircraft are safely separated, maintain high traffic flow, and accommodate operator preferences when possible, will be challenging at UML-4. Functions provided by FPM will be critical at such a maturity level.

UML-4 is expected to rely on *collaborative and responsible* automation, defined in Ref. [4] as automation “assured to perform specified functions such that human monitoring and mitigation of potential failures of those functions is no longer necessary.” The onboard automation systems must account for many constraints of UAM operations, including but not limited to the following:

- Limited endurance of eVTOL aircraft, requiring ultra-efficient flight paths.
- Dynamic scheduling of high-demand vertiports, requiring reliable on-time arrival.
- Geographic constraints defined by municipalities, air traffic control (ATC), and airport traffic flows.
- Altitude constraints defined by buildings, terrain, ATC, and airport traffic flows.
- Coordination procedures that consume decision-making time.
- Small separation requirements to enable sufficient traffic density and volume.
- Wide-ranging aircraft performance of varying UAM aircraft configurations.
- Interactions with traffic operating under different flight rules.

Moreover, the effects of these constraints are likely to change during flight as new flights are planned, predicted conditions change, or emergency airspace closures occur.

C. Intended Function

The objective of an FPM automation system is to create and maintain a flight path that is feasible, deconflicted, harmonized, flexible, and optimal [5].

- A *feasible* path is one that conforms to the aircraft performance and range capabilities; complies with the airspace structure, rules, and constraints; avoids the terrain and charted obstacles; and meets the arrival constraints.
- A *deconflicted* path is one that avoids unsafe proximity to known aircraft, dynamic obstacles, inclement weather, and other emergent airspace hazards.
- A *harmonized* path is one that follows cooperative rules and procedures to ensure that the use of the airspace is coordinated with other airspace users.
- A *flexible* path is one that retains adequate maneuvering room to ensure that future flight path changes, if needed, are available and feasible.
- An *optimal* path is one that best achieves the operator's business objectives for the specific flight.

The primary tasks of an FPM automation system are

- to create the flight path,
- to monitor the flight path and the factors which may impact it,
- to evaluate ongoing acceptability of the flight path and changes proposed by FPM, the air crew, or other parties,
- to revise the flight path, as needed, to sustain the desired qualities, and
- to coordinate the flight path with other airspace users and service providers.

The computed flight path accounts for mission objectives, pilot inputs, aircraft performance, ownship and traffic flight information, current and predicted atmospheric conditions, obstacles and terrain constraints, community rules, and variations in airspace and aerodrome configurations. The FPM automation system generates a four-dimensional (4D) flight path, associated operational intent, and an energy consumption profile, as well as contextual display information, flight conformance status, system health and alerts, and contingency response options.

III. The Flight Test

The Integration of Automated Systems (IAS) flight test was performed at Sikorsky Memorial Airport in Stratford, Connecticut, USA in October 2023.

A. The Aircraft

In the batch and HITL simulation studies leading up to the flight test, hundreds of aircraft were equipped with AOP and interacted within a metropolitan airspace. Conflicts between aircraft were an emergent property of a simulation scenario, not planned. In order to meaningfully observe how FPM dealt with conflicts and other events in a flight test, it was necessary to plan these events to occur to a live aircraft. AOP will usually require only one aircraft to maneuver in response to such an event, so each test point was set up to force one live aircraft, designated the *ownship*, to maneuver. When the maneuver was to be caused by a conflict, a second live aircraft, the *intruder*, provided the conflicting flight plan.

For the flight test, AOP was integrated into computing systems on board the subject aircraft, the Sikorsky Autonomy Research Aircraft (SARA), using additional software developed by the NASA Armstrong Flight Research Center (AFRC) [2]. SARA was designated as the ownship in all test points. Lockheed Martin Sikorsky's S-70™[*] Optionally Piloted Vehicle (OPV) performed the role of the live intruder when required. These two aircraft are shown in Fig. 1. Both aircraft had human pilots. To represent the large number of additional aircraft with which FPM functions would need to avoid conflicts in UML-4 operations, states and flight plans for virtual "background" aircraft were injected into the data stream aboard SARA.

Since SARA was the ownship in each test point, only SARA was equipped with AOP. To supply suitable physical intruder aircraft trajectories for the flight test, an AOP-based scenario generator also produced 4D flight plans to be flown by OPV, which were uploaded to the vehicle during the flight test. We are not aware of any limitations that would have prevented OPV from hosting and using AOP itself, but such an exercise was considered beyond the scope of the flight test.

---

[*] S-70 is a trademark of Lockheed Martin Corporation.

(a) SARA, the ownship. **(b) OPV, the intruder.**

Fig. 1 Aircraft used in the flight test.
Photo credit: Lockheed Martin Sikorsky Aircraft. Used with permission.

B. Additional Computing Hardware

A tablet hosted the user interface for the research pilot aboard SARA during the flight test. AOP itself was hosted on an Intel NUC11TNHi7 Next Unit of Computing (NUC) installed on SARA. The NUC had 64 GB of memory and a processor with 4 cores, 8 hardware threads, and a maximum clock rate of 4.7 GHz. It ran a Windows operating system.

C. Test Points

The test points for the flight test included conflicts between SARA and OPV for SARA to resolve, modified RTAs for SARA to meet, and modified RTAs for SARA that put it in conflict with OPV. A total of 52 test points were developed, divided into eight groups (Group 1 through Group 8), each of which was intended to address a distinct subset of the flight test's research goals [2]. During the flight test, 34 test points were tested to completion in actual flight. In addition to the 34 completed test points, a few attempted tests had to be abandoned because SARA or OPV or both aircraft were unable to set up the desired test conditions. In 6 of these incomplete test runs, AOP collected data relevant to the performance of the aircraft in following the 4D flight plan. Descriptions of the test points are provided in Ref. [2].

IV. Adapting the Autonomous Operations Planner

Autonomous Operations Planner [6] is software developed by NASA as a prototype FPM automation system for research into decision-making in future civil aviation operations. An independent instance of AOP was conceived to reside on the flight deck of each participating aircraft in its operational environment, providing advisories for that aircraft (the ownship to that instance of AOP). Previous versions of AOP explored self-separation concepts for commercial transport aircraft in en route airspace [7], mixed operations with Instrument Flight Rules (IFR) aircraft under traditional air traffic control (ATC) [8], the Dynamic Multi-Track Airway (DMA) concept [9], the Traffic Aware Strategic Aircrew Requests (TASAR) concept advising aircrews flying under IFR and ATC of beneficial route modifications [10], and Trajectory-Based Operations [11].

The purpose of the AOP research prototype is to serve as a reference capability, to establish concept feasibility, to assess the state of technology maturity, and to support development of functional requirements for future operational systems. The AOP prototype demonstrates some of the functions that are required for the "collaborative and responsible" automation required by UML-4. (As a reference prototype, AOP does not provide the level of software assurance implied by this requirement.)

Table 1 lists some salient features of AOP according to how they relate to the five critical properties of a vehicle flight path within the FPM concept. Each of these features is inherited from versions of AOP used in previous efforts and is in either active or anticipated use in other ongoing work, in addition to the IAS flight test.

The remainder of this section describes how we added or modified functions of AOP to adapt it to UAM operations and to integrate it into the IAS flight test.

Table 1 Summary of AOP implementation of the FPM concept for IAS.

Property	Implementation in AOP
Feasible	Generated trajectories use a model of the ownship's performance capabilities and operating limits. Trajectory generation accounts for the ownship's state, including energy on board. Generated trajectories comply with crossing restrictions, special use airspace restrictions, and instructions to arrive at a required time. Constraints can be relaxed in order to satisfy higher-priority constraints.
Deconflicted	*Conflict resolution* produces deconflicted trajectories that avoid traffic, current and forecast 4D convective weather, and 4D restricted airspace. Multiple conflict resolution alternatives are identified to meet replanning objectives, as available. Alternatives are offered for lateral path changes, vertical path changes, a combination of lateral and vertical changes, and speed changes.
Harmonized	Conflict resolution maneuvers are coordinated with traffic using priority rules (right-of-way rules). Resolution maneuvers employ conflict prevention: resolutions will not create new conflicts with other traffic. Trajectories respect changes to an RTA.
Flexible	Not in scope for this effort.
Optimal	Generated trajectories provide paths of lowest energy expenditure, shortest distance, or shortest time (depending on operator preferences).

A. Format for Transmitting Flight Plans

Previous simulation experiments in en route airspace assumed that trajectory intent would be exchanged between aircraft via Automatic Dependent Surveillance-Broadcast (ADS-B), specifically by means of the Trajectory Change (TC) report design that was described in the ADS-B standard but is not supported by any equipment known to us. For UAM operations, the TC reports were replaced by the exchange of flight plans in the Efficient Universal Trajectory Language (EUTL) [12]. This format allows precise definition of a nominal 4D flight plan for aircraft guidance to follow. An example of a 4D flight plan in this format appears in the Appendix.

In order to support the export of flight plans in EUTL format, we enhanced the internal representation of AOP's trajectory change points beyond the requirements for TC reports. We added functions for converting trajectories from the internal AOP format to EUTL and back again. We developed a new version of the component of AOP that tracks traffic aircraft states and trajectories. This component received trajectory intent information in EUTL format rather than in the form of TC reports.

The exchange of flight plans via EUTL does not replace the implemented functions of ADS-B or any other mechanism that transmits the identity and sensed position and velocity of an aircraft.

B. Predicting Trajectories of UAM Aircraft

Planning feasible, optimized flight paths for an operational environment such as UAM requires a function to predict a 4D trajectory (latitude, longitude, altitude, and estimated time of arrival (ETA) at each intermediate point) that accounts for aircraft performance limitations and accurately tracks fuel or energy usage. The trajectory prediction function must follow specified procedures for climb and descent and must be able to meet a required time of arrival.

For the UAM environment, we used the Behavior-Based Trajectory Generator (BBTG) [13], an object-oriented library for intent-based trajectory prediction originally developed for the work described in Ref. [10]. One of the design goals of BBTG was to enable more variety in the types of aircraft, operations, and aircraft performance data than trajectory prediction in AOP previously supported. In BBTG, a trajectory is predicted via a sequence of *atomic* *behavior models* (one of several types of software object defined in the BBTG framework [13]), each determining how the aircraft should fly some part of its flight path, organized hierarchically into *compound behavior models* and supported by *math models* representing the equations of motion under the constraints of the atomic behavior.

This entire structure can be reconfigured by the substitution of variant behavior models at any level, while evaluation functions that depend on aircraft performance data can be interchanged with functions that accept different forms of the performance data. This design provided relatively straightforward paths to implement most of the features required in the UAM environment in general and in the IAS flight test in particular.

1. Vertical Takeoff and Landing

We created new atomic behavior models and math models to model vertical takeoff and landing in BBTG, which no previous BBTG aircraft models had been able to perform.

Predicting these behaviors in a wind field in simulation brought attention to BBTG code that assumed the true airspeed would be greater than wind speed. We eliminated this assumption from the relevant functions.

2. Battery Usage

We added battery powered aircraft to BBTG. The code to support this included support for units of power and electric charge, new fields representing battery charge in existing aircraft-state data structures in AOP, and math models to predict the drawdown of charge under various conditions according to tables loaded from UAM aircraft performance model files.

3. Procedural Aircraft Performance Models

We developed models for two reference UAM aircraft: a quadrotor eVTOL and a lift-cruise tiltrotor eVTOL.

For these models, we generated simple tables of climb rates, descent rates, and other properties as functions of indicated airspeed and altitude by measuring the vehicle capabilities in the simulator and then slightly de-rating the limit performance. This allowed BBTG to produce trajectories near the limit of vehicle performance while reserving some performance margin for the vehicle guidance system to use to maintain 4D conformance in the presence of disturbances and wind prediction errors.

We consider these to be “procedural” aircraft performance models, in contrast to the detailed kinetic performance models (including such things as thrust and drag tables) previously used for almost all aircraft in BBTG. In principle, BBTG could use kinetic performance models of UAM aircraft, but this was not required for the immediate research goals.

4. Acceleration and Deceleration

In previous applications, BBTG predicted a constant airspeed within most behavior models; the only exceptions were behaviors with a cost index, in which the speed could vary over time as the gross weight of the aircraft changed. It was not considered necessary to model the accelerations and decelerations by which an actual aircraft would change airspeeds between these behaviors.

For UAM, we anticipated that the aircraft would spend a much larger portion of its flight accelerating or decelerating along the flight path, so we decided to model explicit accelerating and decelerating behaviors. We created new math models to support these behavior models.

Initially, we selected indicated airspeed, rather than time, altitude, or distance, as the variable of integration for the new math models. After some experimentation, we changed the variable of integration to ground speed to enable the models to function better. Moreover, we found that when a condition is encountered during an integration step that forces the step to end early in order to transition to a new behavior, the interpolation of state data previously used by AOP produced timesteps in which the mean ground speed was inconsistent with the distance and elapsed time. In order to make these data consistent without increasing the processing load more than necessary, we used the general formula for the coefficients of the explicit second-order Runge-Kutta method in the computation of these transitional integration steps.

5. Turn Rates

For the IAS flight test, we modified AOP to keep turn rates within the limits of the integrated guidance systems aboard the flight test aircraft. We added a property to the aircraft performance models enabling them to specify how turns would be computed and we used this property to set the turn rate to 3 degrees per second for UAM vehicles. This enabled the guidance systems to conform to the 4D flight plan better than the existing turn computations, which assumed bank angles more suitable for higher-speed aircraft.

6. Speed Constraints During the Cruise Phase

An enhanced strategic replanning capability developed for the UAM simulation studies (see Subsection IV.E) required BBTG to be able to apply speed constraints during the cruise phase. We also introduced a "hold airspeed" constraint to prevent AOP from planning an immediate speed change in order to meet the RTA despite a new speed constraint. Constraining the speed too much during cruise, however, could prevent AOP from planning a flight path to meet the RTA, so we introduced the "release" speed constraint. Aircraft speed on the portion of a trajectory after a "release" constraint was constrained only by aircraft performance, even if there was an ordinary speed constraint earlier in the trajectory. That is, a "release" constraint canceled any previous constraint, allowing AOP to adjust the planned airspeed after that point.

7. Predicting a Complete Trajectory

The similarities between the overall structures of the trajectories of a UAM aircraft and conventional aircraft allowed the function that generates a complete UAM prediction to reuse much of the code developed for a conventional fixed-wing trajectory. Existing high-level behaviors in BBTG, however, did not always account for various details mentioned above. We developed UAM-specific versions of the behavior models for the climb, cruise, and descent phases of flight as alternatives to the behavior models that supported conventional operations.

C. Conflict Detection

The *conflict detection* (CD) function of AOP determines whether an ownship trajectory leads to a conflict. A *conflict* is a predicted loss of separation between the ownship and another aircraft or a predicted penetration of an airspace hazard. For the UAM operational environment, we reused the existing CD algorithm in AOP, which provides alerts for conflicts while accounting for bounded uncertainties in trajectory prediction [13], substantially as it previously existed, but we reconfigured its parameters to incorporate candidate separation requirements for UML-4. These separation requirements were roughly equivalent to DO365A Detect and Avoid (DAA) Well Clear volumes:

- 1500 feet lateral (approximately 0.25 nautical mile).
- ±450 feet vertical.

We set the Trajectory Prediction Uncertainty Bounds to appropriate values for the flight test. Section V.B provides details of this selection process.

We reduced the lookahead time for alerting to a conflict (the time to go until first loss of separation at which the air crew will be notified of a conflict) from 10 minutes to a baseline of 3 minutes. (Some test points in the flight test set lookahead times to 2 minutes, 4 minutes, or 4.5 minutes.) The reduced lookahead time reflects the relatively short duration of UAM flights and our anticipation of a highly dynamic environment due to high traffic density.

We decreased the CD cycle length (which controls the frequency of recomputing CD results) from 10 seconds to 3 seconds, thereby increasing the frequency of CD computation.

D. Flight Planning in the UAM Domain

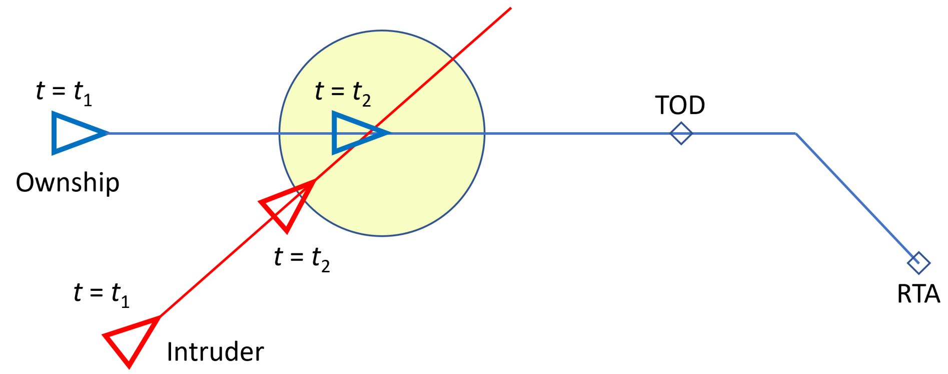
The original design of AOP assumed that a Flight Management System (FMS) would define the trajectory of the aircraft for the guidance system. In order to evaluate a maneuver during strategic replanning, AOP would estimate (as well as it could) the trajectory that the FMS would construct as a result of the maneuver. The predictions made by AOP therefore tended to correspond with the 4D flight plan guided by the FMS during simulation, and these AOP-generated predictions were used in order to create deconflicted trajectories.

In the UAM environment, a suitable model of an FMS was not available. Instead, BBTG produced an initial 4D flight plan that we considered to be reasonable for a UAM flight.

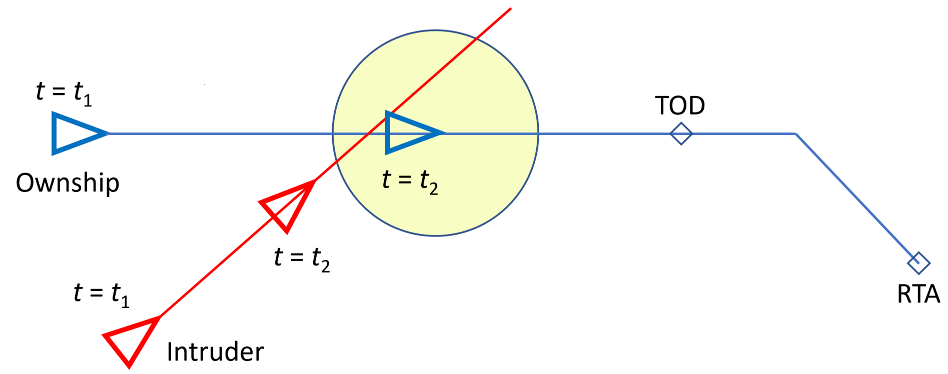
1. Obtaining the Initial Active 4D Flight Plan

We initially used a Trajectory Generator (TG) other than BBTG to initialize the active 4D flight plans of an IAS scenario due to the availability of a scenario generation tool with this capability. The TG used by this tool—which we call the *foreign* TG to distinguish it from BBTG in the following discussion—produced a plausible (though not optimal) flight plan for a UAM vehicle, constructed according to different constraints and assumptions from those of a BBTG trajectory.

In simulations of this design, we found that conflicts caused by an active trajectory from the foreign TG often were eliminated simply by using a new 4D flight plan predicted by BBTG. This is illustrated in Fig. 2, which shows two possible trajectories for an aircraft whose mission plan requires it to arrive at its destination at an assigned RTA. In each part of the figure, the ownship starts at the left side of the diagram at time t_1, flies at the cruise altitude until the top of descent (TOD), then descends to the destination at the lower right of the figure, arriving at the given RTA. Also at time t_1, an intruder aircraft starts flying a course that crosses the ownship's path at the ownship's cruise altitude.



(a) Conflict with flight plan from foreign TG.



(b) No conflict with flight plan from BBTG.

Fig. 2 How conflicts depend on the choice of trajectory generator.

Figure 2a shows the 4D flight plan predicted by the foreign TG, which flies at speeds chosen by that TG to meet the RTA at the destination. According to that trajectory, at time t_2 the ownship is separated from the intruder by less than the minimum separation, indicated by a disk around the ownship's position at t_2; this is a conflict.

Figure 2b shows the 4D trajectory predicted by BBTG, in which the predicted airspeed after the TOD is slower than that predicted by the foreign TG. This requires BBTG to predict a faster cruise speed in order to meet the RTA. Because of the faster cruise speed, the ownship is ahead of the intruder as the intruder approaches the ownship's flight path (at t_2), and there is never a loss of separation.

Since the ownship was guided by the trajectory from the foreign TG in these simulations but the CD algorithm used the BBTG prediction of the trajectory, this discrepancy between the assumptions of BBTG and the foreign TG caused the CD algorithm to erroneously declare the 4D flight plan conflict-free.

Initially, we attempted to solve this problem by forcing the UAM version of AOP to use the EUTL plan received from the aircraft, not an AOP-predicted EUTL plan, when performing CD on the active flight plan. But when AOP was allowed to replan the 4D trajectory in order to resolve the conflict shown in Fig. 2a, using BBTG to predict the 4D trajectory of each proposed resolution maneuver, almost every possible maneuver resulted in a 4D trajectory with a higher cruise speed than the active flight plan, leading to a result similar to Fig. 2b. The optimal resolution maneuver finally chosen by AOP tended to be close to the lateral path and altitude of the original flight plan, the only significant difference being which TG computed the speed along the flight path.

We do not consider such a "resolution maneuver" to be representative of the kind of strategic replanning that would or should be performed in an operational system. For that reason, the initial 4D flight plan, as well as all 4D flight plans created by strategic replanning, needed to be generated by BBTG.

2. Mission Plans

In order to enforce consistency of assumptions and constraints between the 4D flight plans before and after strategic replanning in the final environment used in the preliminary simulation studies, each UAM aircraft received only a *mission plan* from the scenario, consisting of an origin aerodrome, a destination aerodrome, (optionally) an RTA at the destination, a cruise altitude, and (optionally) a list of fly-by waypoints between the origin and destination. The UAM aircraft delivered this mission plan to its on-board AOP, which responded with a EUTL plan (based on a BBTG trajectory) to be used as the active 4D flight plan.

3. Use of Wind in Predictions

Although EUTL describes its 4D trajectory in terms of ground speed, UAM aircraft should be flown at preferred airspeeds when possible and should be constrained by airspeed rather than ground speed during strategic planning and replanning. This requires BBTG to take wind speeds along the flight path into account. We assumed that a useful prediction of wind speed would be available to AOP during the flight and that BBTG would use this prediction.

For the flight test, we determined that AOP and BBTG would need to take into account the wind on the test range. That is, AOP would upload a predicted wind field that approximated the actual wind observed at the time of the test. This introduced another source of speed variations in the 4D flight plan, since BBTG would adjust airspeeds to meet the RTA in the presence of a headwind differently than it would for a tailwind. In order to create conflicts with the desired geometries and other properties for study, it was necessary to take additional steps prior to and during the flight test in order to provide appropriate starting locations for the ownship and intruder, mission plans for the ownship, and EUTL plans for the intruder based on the predicted wind. Ref. [2] provides a description of these efforts.

E. Strategic Trajectory Replanning

Strategic trajectory replanning is a core capability of AOP. This capability generates new optimized flight paths that avoid conflicts with traffic aircraft, regions of special use airspace, regions of severe weather, and other hazards while meeting the requirements of the aircraft's mission such as arrival at a vertiport subject to an RTA. An important use of this capability is the resolution of conflicts that are detected during flight. The Pattern-Based Genetic Algorithm (PBGA) [14, 15] performs strategic trajectory replanning for AOP.

For the UAM operational environment, we added or modified the following capabilities in PBGA:

- We added a "minimum battery usage" objective to the choices of criteria on which to optimize the trajectory. The AOP configuration for the flight test selected this objective.
- We reduced the baseline lookahead time for conflict resolution (the time period during which a maneuver must create no new conflicts) from 20 minutes to 6 minutes.
- We introduced a parameter to configure the resolution *freeze horizon* (the earliest point at which a resolution maneuver may begin), which was previously a fixed value of 15 nautical miles. We enabled the freeze horizon to be specified as either a distance or a time value. For the UAM domain, we configured the freeze horizon to 40 seconds.
- We introduced a separate parameter to configure the time interval between conflict resolution attempts, which previously was determined by the CD cycle length. We configured this interval to 20 seconds in the UAM domain.

- We added a "speed" degree of freedom to the existing degrees of freedom (lateral, vertical, and combined lateral-vertical). We implemented one maneuver pattern for the speed degree of freedom. This maneuver pattern imposes a speed constraint during cruise to alter the time of arrival at the loss-of-separation point. The new maneuver also uses two newly-developed types of speed constraint (see Section IV.B.6). It inserts a "release" speed constraint at a later point in the flight path to enable AOP to plan to meet the RTA. (All flights were assumed to have RTAs during the flight test.) It inserts a "hold airspeed" constraint at the start of the new prediction to set the initial airspeed consistent with the previous airspeed when computing the new 4D flight plan and to hold the airspeed constant until the freeze horizon.
- We enhanced data collection within AOP to capture data desired for post-flight data analysis, such as the exact amounts by which proposed maneuvers within PBGA improved or disimproved battery usage, which was the optimization criterion for the flight test.
- To accommodate UML-4 traffic density levels, we parallelized the implementation of PBGA. This redesign of PBGA takes advantage of the growth in availability of multiple-core processors since the original implementation.

We made various key parameters of PBGA (number of generations, number of individuals in a generation, survival rate, rate of mutation, and number of individuals shielded from mutation) configurable but set them at the levels used in previous experiments. We hope to explore this capability in future work.

F. Tactical Trajectory Replanning

Configurations of AOP from past experiments included multiple layers of tactical conflict resolution, in which the algorithms do not attempt to preserve all mission objectives when resolving conflicts. For IAS, only strategic planning capabilities of AOP were considered; therefore all tactical conflict resolution capabilities in AOP were disabled for the flight test. Instead, the IAS flight test included testing of a separate Hazard Perception and Avoidance (HPA) function.

G. Input and Output

In previous instantiations of AOP for conventional transport flight decks, AOP transmitted and received data to and from other avionics via communications protocols based on ARINC-429 word formats. For UAM, there were no equivalents of the actual or virtual flight-deck components from which AOP received ARINC-429 data. In the UAM simulations and flight test, the aircraft exchanged messages with AOP in the form of C++ data objects that were marshaled and unmarshaled by a relatively simple communications library.

V. Results

A. Computational Speed of AOP

The primary concern about the speed of computation in AOP is the elapsed time required to complete a set of advisories in PBGA. During the simulation studies leading up to the flight test, however, when the number of aircraft in the scenario was about 150 aircraft, the elapsed time of PBGA was sometimes 20 seconds or longer. This was longer than the time allowed for a conflict resolution and caused the resolution to be canceled and any partial results to be discarded. With this motivation, we modified PBGA to perform parallel processing during the simulation studies. PBGA continued to use parallel processing in the flight test, where the number of traffic aircraft was increased.

During the flight test, PBGA used four parallel threads and processed about 250 traffic aircraft (including OPV and background traffic) in each run, except for Group 8 test runs, during which AOP was tracking more than 330 traffic aircraft. The elapsed time from the first detection of a conflict until the presentation of the first complete set of conflict resolution advisories was typically 8.8 seconds. No elapsed times greater than 12.1 seconds were observed [1].

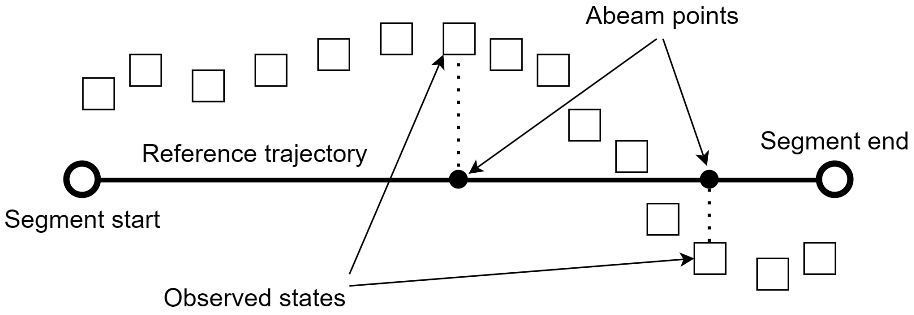
B. Estimating Trajectory Prediction Uncertainty Bounds

As part of the preparation for the flight test, we estimated the Trajectory Prediction Uncertainty Bounds (TPUBs) that should be assigned to each part of each aircraft's trajectory, since the detection by AOP of a conflict between SARA and OPV could be significantly affected by the magnitude of the TPUBs assigned to each part of each aircraft's trajectory.

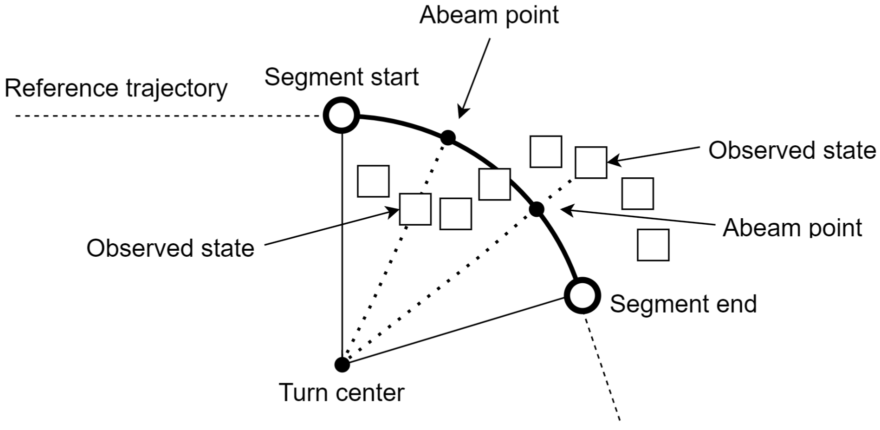
If the TPUBs were set too small, there would be a possibility of false negative results from the CD algorithm. That is, CD could predict that two aircraft would have the required separation based on their predicted flight plans and anticipated deviations from those plans, but a deviation larger than CD allowed for (which we considered to be non-conformance with the 4D flight plan) could cause the aircraft to lose separation during the actual flight. On the other hand, setting the TPUBs to very large values in order to minimize the risk of non-conformance could cause CD to protect too much airspace and could limit the density of operations.

The procedure to estimate appropriate TPUBs values made use of the limited amount of flight time that was allotted prior to the flight test for performance testing of the guidance systems of SARA and OPV in combination with an additional layer of NASA-developed software ("NASA Middleware") that dynamically interpreted the 4D flight plan from AOP. We developed the FPM/AOP TPUBs Calibration Analysis Tool (FATCAT) to identify the relevant sequence of observed states of an aircraft (latitudes, longitudes, and altitudes at recorded times) and to compare the observed states with a 4D flight plan (in the form of a EUTL plan) in the dimensions of altitude deviation, cross-track (lateral) deviation, and along-path/time (longitudinal) deviation.

In order to measure the deviations of observed states from the 4D flight plan, FATCAT partitions the plan into a sequence of straight segments (geodetic paths between waypoints) and turn segments (paths that follow small-circle arcs). FATCAT maps each observed state onto one of these segments. A state mapped onto a straight segment is projected orthogonally onto the segment as shown in Fig. 3a. A state mapped onto a turn segment is projected radially onto the segment as shown in Fig. 3b, using the center of the turn arc as the center of the projection. (The deviations of the observed states from the EUTL plan in Fig. 3 are exaggerated for the purpose of illustration.) The point to which the observed state is projected is its *abeam point*.



(a) On a straight segment.



(b) In a turn segment.

Fig. 3 Abeam points used in measurements by FATCAT.

The altitude deviation of an observed state is the signed difference of the altitude of the observed state and the predicted altitude of the EUTL plan at the abeam point. The time deviation is the signed difference of the recorded time of the observed state and the predicted time at the abeam point according to the EUTL plan. The magnitude of the cross-track deviation is the distance to the abeam point; FATCAT distinguishes between deviations to the left and deviations to the right of the path by the sign of the deviation and makes a separate record of deviations from turn arcs in which the sign indicates whether the observed state is outside the turn arc or inside it.

Following preliminary performance testing of the 4D guidance system installed on both aircraft for the purpose of the IAS flight tests, we set the TPUBs around the predicted trajectories to 150 feet left or right of the nominal path, 25 feet above or below the nominal path, and 3 seconds before or after the ETA at each point. Essentially all observed data points from correctly functioning performance-testing flights were within these bounds.

C. Conformance to Ownship Intent

For the flight test, an aircraft was considered in conformance with its 4D flight plan if the deviations from that plan were within the limits defined by the TPUBs at each point. The instance of AOP aboard SARA collected 4D flight plans and aircraft state data from 34 completed test runs in addition to six incomplete test runs and one ground test. We submitted all the collected ownship data to FATCAT analysis, which compared each ownship state to the relevant 4D flight plan from among the 77 flight plans followed by SARA. (Some test runs followed more than one flight plan due to the execution of advisories during the test run.)

In most runs, altitudes were in conformance and the conformance in the cross-track and along-path dimensions was excellent. Typical maximum lateral deviations were less than 10 feet and typical maximum along-path deviations were less than 1 second.

There were conformance violations in ten runs. These are organized into categories below. Lateral deviations tended to be smaller than vertical deviations although the TPUBs in the lateral dimension were much larger. We attribute this to improvements that were made in NASA Middleware after the final TPUBs configuration was set.

1. Altitude Violations in Level Flight

In two runs, there were two excursions lasting 3 seconds each outside the ±25 feet vertical TPUBs during level flight: 27.3 feet low in one case, and 27.8 feet high in the other. (These figures are measured relative to SARA's sensed altitude and represent flight technical error; the dGPS unit providing the altitude had a published vertical accuracy of 13 feet.) These violations may have been due to transitions in the flight plan that affected the stability of the altitude tracking. They may also indicate that the vertical TPUBs were set too small (at ±25 feet) due to concerns about the effects of altitude deviations, since cruise altitudes were spaced only 500 feet apart and the vertical separation requirement was 450 feet.

2. Altitude Violations in Non-Level Flight

In two runs there were violations of vertical TPUBs during a climb: 44.5 feet below and 88.6 feet below.

3. Cross-Track Violations

In one run there was a cross-track deviation of 432.5 feet. This occurred in the first turn of a lateral conflict resolution advisory. Data recorded from this run indicated that AOP delivered the advisory to the NASA Middleware 34 seconds before the planned beginning of turn (BOT), but AOP received the response from the research pilot approximately 11 seconds after the planned BOT. That is, by the time AOP received the pilot's response, SARA had already overrun the start of the turn. The along-path conformance during this maneuver remained well within the TPUBs bounds of ±3 seconds.

4. Along-Path Violations in Completed Runs

In two completed runs there were along-path conformance violations. In both cases, SARA later recovered conformance; the maximum deviations in those runs were 5.1 seconds early and 3.64 seconds late, respectively.

5. Along-Path Violations in Aborted Runs

In three aborted test runs, SARA was out of conformance immediately after changing speed to meet an RTA, and it never regained conformance. All three test runs were attempts at a single test point that was intended to examine a conflict after the change in the RTA. Due to the conformance failures, this test point was never completed.

D. Stability of Conflict Detection

In preparation for the flight test, we simulated the test scenarios to confirm their expected behaviors. During some of these simulations, in cases where SARA and OPV were planned to fly parallel courses, one aircraft overtaking the other, we observed intermittent conflicts and conflicts for which the location of the loss of separation changed from time to time. Due to schedule constraints, it was not possible to eliminate these behaviors prior to the flight test. The scenarios in which these behaviors occurred remained in the test plan and were flown during the flight test as time permitted.

During the flight test, these behaviors occurred in two runs, one from Group 4 and one from Group 7 according to the nomenclature of the test plan [2]. In the run from Group 4, the conflict disappeared and appeared again, repeatedly. The conflict eventually became stable for a few seconds, AOP generated an advisory, and the research pilot was able to select and execute the advisory, which resolved the conflict. In the run from Group 7, the location of the conflict alternated between two positions on the flight plan, and AOP never generated an advisory; this was consistent with a prior observation of that run in simulation.

In all other cases, AOP consistently detected conflicts where the test scenario intended a conflict to occur, when the scenario intended it to be detected, and the conflict remained at essentially the same location until it was resolved by a maneuver or until the run ended.

VI. Discussion and Recommendations

A. Ability to Conform to a 4D Path

In Section V.B we described how we used guidance performance tests prior to the flight test in order to set the TPUBs values that would be used. By the time of the flight test itself, the NASA Middleware had been fine-tuned to a point where the aircraft followed the 4D path much more closely in both the cross-track and along-path (time) dimensions than they had during the procedure for setting TPUBs. At first glance, it appears that some of the conformance limits could be made much tighter and still be achievable in practice, especially if a few anomalies are addressed as discussed later in this paper.

The results of the flight test suggest to us that tight conformance to a 4D flight plan is *possible*. It is important to ask whether such tight conformance is *desirable*. It may be that a “gentler” guidance system that allowed larger deviations from the 4D flight plan would provide better ride quality or better energy efficiency.

Just as large separation requirements could restrict traffic density, very large TPUBs could require AOP to protect airspace far greater than the given separation requirements and over-constrain planning in a dense airspace. But lateral TPUBs of ±150 feet, much greater than the typical lateral deviation observed in the flight test, do not seem to cause any difficulty in the UAM environment. Similarly, re-running the preliminary simulation experiments with along-path TPUBs of ±12 seconds (four times what the flight test used) had no apparent effect on AOP performance.

Not surprisingly, altitude conformance appears to be better during level flight than during climb. The two minor violations of conformance in level flight appeared to be due to speed transitions. Although we set vertical TPUBs to a constant ±25 feet for the flight test, the TPUBs design in AOP does not require any of the bounds to be constant over any part of a trajectory, and the vertical TPUBs can be asymmetric above and below the predicted flight path. Extending the vertical TPUBs to 100 feet above or below the path during parts of climbs and descents could easily have covered all vertical deviations observed during non-level flight in the flight test while making only a minimal reduction of the airspace that could be used by other aircraft.

B. Stability of Conflict Detection

Prior to the flight test, we became concerned about the fact that AOP repredicts the 4D trajectories of both the ownship and all traffic aircraft prior to each execution of the CD algorithm. These repredictions originally were motivated by the assumption that the ownship trajectory was merely AOP’s attempt to reproduce the 4D guidance generated by an FMS and that the traffic trajectory was received in the form of TC reports in ADS-B, which are not as descriptive as EUTL and require the use of state vector reports to infer the first part of the trajectory.

When CD is performed under these circumstances, with frequent changes in the predicted trajectory of each aircraft and no regard for previous CD results, there will necessarily be “edge cases” in which the data at one time indicate a possible future loss of separation and the (modified) data a few seconds later do not, or vice versa, due to input data that at various times either barely pass or barely do not pass whatever threshold is set for the detection of a conflict. Similarly, there will be cases in which the location of first loss of separation of a detected conflict will change dramatically from time to time.

Since AOP repredicted trajectories continually during the simulation studies and the flight test, it was not surprising that we observed instability of CD in certain circumstances in simulation. We observed the same behaviors again in the same circumstances during the flight test.

For the UAM environment, we propose that the 4D trajectories shared by aircraft should be regarded as contracts those aircraft will follow (within some limits of conformance) until the contracts are (possibly unilaterally) revoked. In that light, AOP should use only the active EUTL plan as the ownship prediction for CD, using the current ownship state only to determine whether that EUTL plan is still valid, and should assume that the EUTL plans of all traffic aircraft are similarly reliable. This may be expected to make the results of CD much more stable, since the exact same input data will produce the exact same results.

C. Quality of Strategic Replanning by AOP

Prior to the flight test, we had identified adverse behaviors attributed to the implementation of strategic replanning in AOP, some of which stemmed from problematic assumptions inherited from earlier versions of AOP developed for an FMS-equipped environment. These behaviors were left unaddressed due to schedule constraints and were replicated in the flight test. The flight test was also an opportunity to comb the data for evidence of previously unidentified defects.

1. Prediction of Speed Advisories for New RTAs

A known defect was that the initial speed advisory made by AOP in response to a change in the RTA assumed an instant change in speed would have occurred at the time when the aircraft received the new RTA. Due to processing time and pilot reaction time, in actual flight the speed advisory was always implemented after a few seconds' delay. In addition, the actual aircraft required some time to accelerate or decelerate to its new airspeed. Due to the failure of the 4D flight plan to allow for this delayed implementation and the period of acceleration or deceleration, by the time the aircraft reached its new airspeed it had already gained or lost time relative to the 4D flight plan.

This problem did not occur in the FMS-equipped environment, where it was the responsibility of the FMS to react to an RTA and where AOP never issued speed advisories.

We attribute the along-path/time non-conformance of SARA in one test point, as described in Subsection V.C.5, to this effect. We observed a less extreme form of the same effect in other test points, but the time deviations in those cases were within conformance and did not interfere with the flight test. We will investigate the use of speed constraints to repair this defect, similarly to the way the speed degree of freedom is designed to avoid an instant speed change.

2. Conflicts Caused by Speed Advisories

The desired function of AOP in response to a change in an RTA in the absence of an FMS is to create a 4D trajectory for a speed advisory, then perform CD on that trajectory and (if necessary) use PBGA to create a new 4D trajectory that is conflict-free. At the time of the flight test, the second half of this function was not yet implemented, and AOP sometimes issued speed advisories that caused SARA to be in conflict with OPV immediately after the pilot executed the advisory.

We were aware of this defect in AOP before the flight test, and we developed a workaround for it: the pilot would wait for PBGA to deliver an additional advisory to resolve the conflict, then execute that advisory. This workaround was incorporated into the pilot instructions for some test points, including the test point from Group 7 described in Subsection V.D. Immediately after the pilot executed the speed advisory for the RTA in this test point, AOP predicted that SARA would lose separation with OPV within 40 seconds. Because we had configured the freeze horizon to 40 seconds, PBGA was forbidden to initiate any maneuver prior to this loss of separation.

While AOP momentarily predicted longer times until first loss of separation at other times after the execution of the speed advisory, the proximity to the conflict may explain why AOP never offered a resolution for this conflict. This was the only test point in which AOP did not provide a suitable conflict-resolution advisory when it detected a conflict in the active 4D flight plan. It provides motivation to repair this known defect in AOP's handling of RTA changes.

3. Speed Changes Due to Conflict Resolution

During the development of the speed maneuver for conflict resolution, we anticipated the problem described in the previous subsection and designed a solution: the speed maneuver should hold the previously planned airspeed until the freeze horizon configured in PBGA, and then accelerate or decelerate to the speed required to avoid a loss of separation.

After the flight test, we discovered that in cases where the aircraft was not already flying near its nominal cruise airspeed, the 4D flight plan generated for this maneuver began with an acceleration or deceleration toward the nominal airspeed, starting at the time when conflict resolution was requested. Like the similar assumption in a speed advisory for a new RTA, this put the aircraft ahead or behind time when the maneuver was finally executed, although the effect was ameliorated by the acceleration or deceleration segment and by the fact that the aircraft was predicted to fly at the nominal airspeed, not the "resolution" airspeed, during this part of the trajectory. We also discovered that other conflict resolution maneuvers could result in advisories that assumed an instantaneous change in speed at the time when PBGA started. For example, when a lateral maneuver made a large reduction in the distance to be flown, PBGA was forced to decrease the cruise airspeed in order to meet the RTA.

None of the conflict-resolution advisories caused SARA to lose along-path/time conformance with the 4D trajectory. We discovered the behaviors described in the previous paragraph because the observed time deviations throughout the flight test were typically less than one second and we examined every larger deviation individually.

4. Continued Applicability of Advisories

The overrun of an initial turn of a maneuver, as described in Subsection V.C.3, indicates a need for AOP to better track the applicability of its advisories as the state of the aircraft changes. In the work of Ref. [10], the Trajectory Aware Planner (TAP) engine performed this function by constantly modifying and repredicting each advisory as the aircraft changed position. The algorithm for doing this did not translate to the maneuvers used in IAS. Instead, the freeze horizon was intended to provide enough time for a pilot to execute an advisory while it was still valid. This design appears to have produced acceptable results in 33 of the 34 test points completed in the flight test but was inadequate in the case described in Subsection V.C.3.

D. Adaptability of AOP

For the most part, adaptation of AOP to the UAM domain and to the particular requirements of the flight test was relatively straightforward due to the modular architecture of AOP itself and of most of the affected components (for example, BBTG).

The adaptation of AOP from a flight deck with an FMS to an aircraft with no FMS brought to light a number of properties of the aircraft's avionics that can fundamentally affect the design of a subsystem. If AOP is to continue to be used in environments (such as the IAS flight test environment) that fundamentally differ from the one in which it was initially implemented, it may be useful to re-examine the design of AOP and either change configuration parameters or implement new versions of some of the components of AOP that are designed for a new environment.

On the other hand, since AOP was originally designed to work in an environment where conformance to a 4D flight plan was not even defined, it already has features designed for such an environment that could be used as needed.

For other possible applications of AOP, this experience indicates that the adaptation to the environment needs to be an early step in the design process, and that all aspects of AOP's interaction with its environment need to be considered. In particular, the amount and type of information available to AOP from other avionics is a critical property of any operating environment, as are the guidance capabilities of any aircraft for which AOP is to generate advisories.

E. Summary

Continued development of AOP may need to address some of the concerns raised during the flight test, including the observed defects in the implementation. We plan to investigate the criteria for updating predicted trajectories as discussed in Subsection VI.B in order to achieve more stable conflict prediction. Better use of speed constraints could improve the continuity of speed guidance resulting from AOP's advisories.

The sensitivity of AOP's CD algorithm in the case of a slowly overtaking aircraft is a topic to investigate. Another question is how small TPUBs need to be (or ought to be) set under any given set of conditions.

Parallel processing enabled PBGA to easily handle as many traffic aircraft as our scenarios could generate. If the speed of execution of PBGA becomes a concern again, the algorithm appears to be well-suited to parallelization over a much larger number of threads than the four-thread execution implemented for IAS. The prevalence of low-cost processors with more than four cores may make this attractive. Faster processing could support possible needs for shorter reaction times, greater exploration of the solution space, or other developments.

It would be beneficial to re-examine the assumptions in AOP's design that have been inherited from the initial plans for using AOP on an FMS-equipped flight deck. An important question is what kind of avionics other than AOP may be expected to be found on aircraft in the future as new markets and concepts of operation emerge. The types and quality of available real-time data, the guidance functions, and other functions that will be available in this future environment will likely have a large impact on the desired functionality of an FPM system. The allocation of functions and responsibilities among components of FPM and associated systems on flight decks or on the ground is a vital concern. We hope to develop a functional architecture that aligns with trends in the development of airborne and ground-based systems while providing the flexibility to adapt to novel ideas, which will enable AOP to provide an enduring research capability.

VII. Conclusion

The observations presented here are based on analysis of the flight test data during and immediately after the flight test, so they may be regarded as preliminary results. These results are promising, as the essential algorithms of AOP performed well. Parallel processing enabled AOP to scale up easily to the complex airspace anticipated for UML-4. There appears to be a clear path to resolve each software defect that was observed.

The UAM simulations and flight test were the first environment in which AOP assumed that aircraft would conform to true 4D flight plans. Our experience in this environment showed that flight plans based on explicit ground speeds, such as EUTL flight plans, are an effective way to communicate aircraft intent to other aircraft (and by extension, to ground systems), even while the flight planning remains based on the airspeed of the aircraft rather than on a fixed ground speed.

The aircraft used in the flight test were able to follow the 4D flight plans from AOP with a high degree of accuracy. The question of what kind of 4D conformance *can* be achieved is now secondary to the question of what kind of conformance *should* be sought.

The UAM simulation environment appears to have been very effective in predicting how the system would operate in actual flight. Even the instability of conflict detection observed in two of the test points had already been observed in simulation.

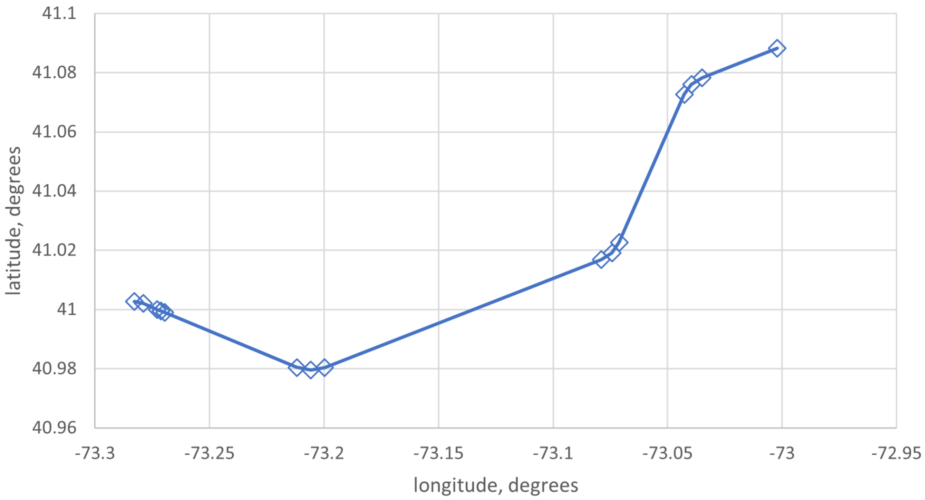
In its extension from an en route, FMS-based environment to the UAM domain, AOP has shown itself to be a flexible and general tool for research into the kinds of advanced flight deck automation that will likely be required in future civil aviation operations. The trajectory prediction algorithm in AOP is easily reconfigured, and the algorithms for conflict detection and conflict resolution translated to the new domain with little more than adjustments to their configuration parameters. At the same time, the flight test suggested possibilities for further advances in the functionality of AOP while bringing attention to the link between the characteristics of future flight decks and the desired functional architecture of AOP.

Over the long term, AOP could be useful in the exploration of emerging aviation markets as well as traditional modes of air transportation. We anticipate that it could support future distributed-agent concepts of operation, whether they involve human operators or are fully automated.

Appendix

Subsection IV.A describes how AOP used the Efficient Universal Trajectory Language (EUTL) [12] to exchange 4D flight plans in the simulation studies and flight test.

Figure 4 shows an example of a EUTL plan generated by AOP for a conflict resolution advisory during simulation of a flight-test scenario. Diamond markers on the map view show the locations of the trajectory change points (TCPs) listed in the encoding of the plan. The encoding lists one TCP per line, specifying the latitude, longitude, and absolute time (in seconds since Jan. 1, 1970, 0000 UTC) at each TCP. The remainder of each line describes features such as the geometry of a turn, including markers for the beginning of turn (BOT) and end of turn (EOT) along with a specified turn center and radius; longitudinal accelerations in order to change ground speed, beginning at BGS and ending at EGS, along with a specified amount of acceleration; and vertical accelerations in order to change vertical rate, beginning at BVS and ending at EVS, along with a specified amount of acceleration. The ground speed at any point between TCPs can be determined from the distance between the TCPs, the time difference between the TCPs, and the longitudinal acceleration, if any, that is defined between those TCPs.



(a) Map of the plan.

```
ID,lat,lon,alt,time,type,tcp_trk,tcp_gs,tcp_vs,radius,accel_gs,accel_vs,center_lat,center_lon,center_alt,info,name
unitless,deg,deg,ft,s,unitless,unitless,unitless,unitless,NM,m/s^2,m/s^2,deg,deg,ft,unitless,unitless
AOP110L,41.08814885,-73.00212920,2000.00000000,1600169062.90000010,Orig,NONE,NONE,NONE,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,41.07825540,-73.03501905,2000.00000000,1600169114.03785062,Orig,BOT,BGS,NONE,-0.63791351,-0.74495200,0.00000000,41.06839911,-73.02979903,609.60000000,-,-
AOP110L,41.07599900,-73.03962510,2000.00000000,1600169122.51021338,Orig,MOT,EGSBGS,NONE,0.00000000,-0.88216260,0.00000000,-,-,-,-,-
AOP110L,41.07258551,-73.04273473,2000.00000000,1600169132.37732053,Orig,EOT,EGS,NONE,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,41.02257153,-73.07118438,2000.00000000,1600169273.08290076,Orig,BOT,BGS,NONE,0.63783975,0.88204201,0.00000000,41.02675282,-73.08411222,609.60000000,-,-
AOP110L,41.01915759,-73.07428948,2000.00000000,1600169282.95052576,Orig,MOT,EGSBGS,NONE,0.00000000,0.74524741,0.00000000,-,-,-,-,-
AOP110L,41.01689988,-73.07888952,2000.00000000,1600169291.42301226,Orig,EOT,EGS,NONE,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,40.98033378,-73.19980732,2000.00000000,1600169479.82579756,Orig,BOT,BGS,NONE,0.72022116,0.25106029,0.00000000,40.99145309,-73.20572181,609.60000000,-,-
AOP110L,40.97947162,-73.20589393,2000.00000000,1600169488.67776299,Orig,MOT,EGSBGS,NONE,0.00000000,-0.27908704,0.00000000,-,-,-,-,-
AOP110L,40.98043321,-73.21195435,2000.00000000,1600169497.54954100,Orig,EOT,EGS,NONE,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,40.99897618,-73.26955271,2000.00000000,1600169588.53214574,Orig,NONE,NONE,BVS,0.00000000,0.00000000,-0.98000000,-,-,-,-,-
AOP110L,40.99950399,-73.27119400,1989.20068064,1600169591.12398243,Orig,NONE,BGS,NONE,0.00000000,0.12842323,0.00000000,-,-,-,-,-
AOP110L,41.00003682,-73.27285096,1956.80271725,1600169593.71581936,Orig,NONE,NONE,EVS,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,41.00198588,-73.27891297,1800.82463150,1600169603.07450438,Orig,BOT,EGSBGS,NONE,-0.47967828,0.12842323,0.00000000,40.99464389,-73.28305669,518.84597604,-,-
AOP110L,41.00262381,-73.28295138,1702.25057756,1600169608.98894763,Orig,MOT,EGS,NONE,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,41.00262113,-73.28334991,1692.96887645,1600169609.54584980,Orig,NONE,BGS,NONE,0.00000000,-1.01149620,0.00000000,-,-,-,-,-
AOP110L,41.00204672,-73.28700582,1601.72863607,1600169615.08028579,Orig,EOT,EGSBGS,NONE,0.00000000,-0.98032533,0.00000000,-,-,-,-,-
AOP110L,41.00030807,-73.29272688,1427.31321310,1600169625.55015039,Orig,NONE,NONE,BVS,0.00000000,0.00000000,0.98000000,-,-,-,-,-
AOP110L,41.00023421,-73.29296988,1419.40034214,1600169626.04917860,Orig,NONE,EGSBGS,NONE,0.00000000,-0.97715360,0.00000000,-,-,-,-,-
AOP110L,41.00016142,-73.29320936,1412.28815528,1600169626.54820681,Orig,NONE,NONE,EVS,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,40.99695935,-73.30374240,823.72237610,1600169670.30829620,Orig,NONE,NONE,BVS,0.00000000,0.00000000,-0.98000000,-,-,-,-,-
AOP110L,40.99695910,-73.30374322,818.80181597,1600169670.65940666,Orig,NONE,EGS,NONE,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,40.99695910,-73.30374322,813.48488784,1600169671.01051712,Orig,NONE,NONE,EVS,0.00000000,0.00000000,0.00000000,-,-,-,-,-
AOP110L,40.99695910,-73.30374322,807.36896317,1600169671.39987731,Orig,NONE,NONE,BVS,0.00000000,0.00000000,0.98000000,-,-,-,-,-
AOP110L,40.99695910,-73.30374322,769.00000000,1600169676.28526998,Orig,NONE,NONE,EVS,0.00000000,0.00000000,0.00000000,-,-,-,-,-
```

(b) Encoding of the plan.

Fig. 4 A sample EUTL plan produced by AOP.

This example was taken from a simulation with a 30-knot wind from 150 degrees true heading. Since indicated airspeed is intended to be held constant during cruise, the wind causes significant changes in ground speed during the turns. There are also ground-speed changes during the descent to the destination. After reducing the ground speed to zero and initiating a vertical descent, the trajectory ends by accelerating vertically to zero vertical speed for a soft touchdown.

Acknowledgments

The authors thank the members of the ATOS development team, the IAS integration team, the flight test team (including personnel from NASA AFRC and Lockheed Martin Sikorsky Aircraft), and the NASA National Campaign who performed or facilitated the development, software integration, and flight testing of the system under study.

References

[1] Barrows, B. A., Ballin, M. G., Barney, T. L., Nelson, S. L., Underwood, M. C., and Wing, D. J., "Sim to Flight: Evaluating Flight Path Management Automation in High Density Urban Environments," *AIAA SCITECH 2024*, January 2024. In press.

[2] Nelson, S. L., Ballin, M. G., Barrows, B. A., Underwood, M. C., Wing, D. J., Sturdy, J. L., and Williams, E. R., "Designing a Flight Test of a Flight Path Management System for Advanced Air Mobility Research," *AIAA SCITECH 2024*, January 2024. In press.

[3] Wing, D. J., Chancey, E. T., Politowicz, M. S., and Ballin, M. G., "Achieving resilient in-flight performance for advanced air mobility through simplified vehicle operations," *AIAA Paper 2020–2915*, June 2020. https://doi.org/10.2514/6.2020-2915.

[4] Goodrich, K. H., and Theodore, C. R., "Description of the NASA Urban Air Mobility Maturity Level (UML) Scale," *AIAA Paper 2021–1627*, January 2021. https://doi.org/10.2514/6.2021-1627.

[5] Karr, D. A., Wing, D. J., Barney, T. L., Sharma, V., Etherington, T. J., , and Sturdy, J. L., "Initial Design Guidelines for Onboard Automation of Flight Path Management," *AIAA Paper 2021–2326*, August 2021. https://doi.org/10.2514/6.2021-2326.

[6] Karr, D. A., Vivona, R. A., Roscoe, D. A., Depascale, S. M., and Wing, D. J., "Autonomous operations planner: A flexible platform for research in flight-deck support for airborne self-separation," *AIAA Paper 2012–5417*, September 2012. https://doi.org/10.2514/6.2012-5417.

[7] Finkelsztein, D., Lung, T.-K., Vivona, R. A., Bunnell, J., Mielke, D., and Chung, W., "Airspace and Traffic Operations Simulation for Distributed ATM Research and Development," *AIAA Paper 2005–6488*, August 2005. https://doi.org/10.2514/6.2005-6488.

[8] Barhydt, R., Kopardekar, P., Battiste, V., Doble, N., Johnson, W., Lee, P., Prevot, T., and Smith, N. T., "Joint NASA Ames/Langley Experimental Evaluation of Integrated Air/Ground Operations for En Route Free Maneuvering," *6th USA/Europe Seminar on Air Traffic Management Research and Development*, Baltimore, MD, June 2005.

[9] Wing, D. J., Smith, J. C., and Ballin, M. G., "Analysis of a Dynamic Multi-Track Airway Concept for Air Traffic Management," NASA TP–2008-215323, July 2008.

[10] Wing, D. J., "The TASAR Project: Launching Aviation on an Optimized Route Toward Aircraft Autonomy," NASA TP–2019-220432, December 2019.

[11] Barney, T. L., Ballard, K. M., Buck, B. K., Underwood, M. C., , and Chartrand, R. C., "Simulation of a Representative Future Trajectory-Based Operations Environment," NASA TM–2020-5006885, November 2020.

[12] Hagen, G. E., Guerreiro, N. M., Maddalon, J. M., and Butler, R. W., "An Efficient Universal Trajectory Language," NASA TM–2017-219669, September 2017.

[13] Karr, D. A., and Vivona, R. A., "Conflict Detection Using Variable Four-Dimensional Uncertainty Bounds to Control Missed Alerts," *AIAA Paper 2006–6057*, August 2006. https://doi.org/10.2514/6.2006-6057.

[14] Vivona, R. A., Karr, D. A., and Roscoe, D. A., "Pattern-Based Genetic Algorithm for Airborne Conflict Resolution," *AIAA Paper 2006–6060*, August 2006. https://doi.org/10.2514/6.2006-6060.

[15] Karr, D. A., Vivona, R. A., Roscoe, D. A., DePascale, S. M., and Consiglio, M., "Experimental Performance of a Genetic Algorithm for Airborne Strategic Conflict Resolution," *AIAA Paper 2009–5748*, August 2009. https://doi.org/10.2514/6.2009-5748.