2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) | 978-1-6654-3420-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/DASC52595.2021.9594493

SIRIUS: Simulation Infrastructure for Research on Interoperating Unmanned Systems

Andrew Peters, Brendan Duffy,
Swee Balachandran
National Institute of Aerospace,
Hampton, VA, 23666
{andrew.l.peters,brendan.j.duffy,
sweewarman.balachandran}@nasa.gov

María Consiglio
NASA Langley Research Center,
Hampton, VA, 23666
maria.c.consiglio@nasa.gov

César Muñoz
NASA Langley Research Center,
Hampton, VA, 23666
cesar.a.munoz@nasa.gov

***Abstract*—This paper presents the Simulation Infrastructure for Research on Interoperating Unmanned Systems (SIRIUS), a research framework for simulation and analysis of future conceptual Urban Air Mobility (UAM) operations. SIRIUS is being developed under the auspices of the NASA Air Traffic Management eXploration project, UAM subproject (ATM-X UAM). SIRIUS provides an intuitive, highly configurable graphical user interface to design complex traffic scenarios and airspace configurations representative of conceptual UAM operations. Aircraft simulated with SIRIUS can be equipped with flight-tested capabilities for detect and avoid (DAA), geofencing, distributed merging and spacing, path conformance, and path planning while executing time-constrained, 4D trajectories generated by a UAM ground operations system. Central to the design of the SIRIUS simulation framework is the capability to evaluate the integration and interoperability of ground-based separation services (e.g., strategic separation) with extended DAA functionality (e.g., path monitoring, separation provision, merging and spacing, etc.) The simulation environment also supports modelling of wind, navigation, and sensor uncertainties, as well as communication delays. SIRIUS enables distributed simulation of large-scale scenarios. An interactive graphical analysis capability helps isolate, visualize, and compare relevant vehicle state data and widely used measures of performance metrics across multiple scenarios.**

Keywords—Urban Air Mobility, Simulation, Monte Carlo Studies, UAM Traffic Management, ICAROUS, DAIDALUS

I. INTRODUCTION

While ongoing efforts to develop initial concepts of operations for the integration of UAM operations in the NAS continue to evolve, a set of preliminary concept papers currently form the foundation for the community's understanding of UAM [1, 2, 3]. These concepts rely on the use of new aircraft technologies and airspace structures supported by highly automated systems to safely manage an expected high demand for UAM operations.

What approach will represent the least disruption to the air traffic system, be acceptable to Air Traffic Control (ATC) and other users of the airspace while preserving the current level of safety in the NAS? What approach will achieve those goals and ensure fair access to the airspace and be scalable? Understanding the impact to safety and to the normal operations of the air traffic system as well as the nature and magnitude of effort that needs to be done is central to UAM integration research. Establishing unambiguous separation responsibilities, as well as development and evaluation of the procedures and prototype technologies necessary to support these new operations is one of the most critical challenges. The technology challenges and integration complexities of such operational concepts will require a phased, incremental approach to ensure both safety and fair access to the airspace for new and existing users.

The National Aeronautics and Space Administration (NASA) is collaborating with the Federal Aviation Administration (FAA) and multiple industry partners to address some of these challenges and facilitate a path for UAM integration in the National Airspace System (NAS). As part of this effort, the NASA UAM subproject of the Air Traffic Management eXploration (ATM-X) project is developing a research implementation roadmap, building software prototypes, and simulation infrastructure to investigate key airspace integration problems ranging from the initial, early, low-demand to latter-day, high demand, high density operations. Addressing the scalability of these operations as well as their interoperability with exiting systems and regulations in the NAS, is most critical, since it is expected that at certain levels of UAM demand, current Air Traffic Control (ATC) operators and systems may be overwhelmed. Hence, new separation and conflict management approaches need to be researched and new performance-based standards for these operations need to be developed.

This paper describes the SIRIUS software application framework for simulation and analysis conflict management concepts in UAM operations. SIRIUS seamlessly integrates on-board flight-tested capabilities for detect and avoid, geofencing, distributed merging and spacing, path conformance, and path planning, as well as ground-based systems for traffic surveillance, flight plan changes, and dynamic constraint updates into a batch simulation environment with an interactive graphical analysis tool.

The rest of the paper is organized as follows: Section II references related tools and research efforts. Section III provides information about relevant systems used within SIRIUS. Section IV provides a detailed description of the SIRIUS architecture and its various components. Section V provides examples of simulations done with SIRIUS. Section VI provides a discussion on current work towards improving SIRIUS, future research directions and concluding remarks.

II. BACKGROUND

Simulation tools play a major role in the design and development of advanced air traffic management concepts. For example, the Single European Sky Air Traffic Management Research (SESAR) project and the Next Generation Air Transportation Systems project both have made use of numerous simulation capabilities to investigate novel design concepts [4,5], 4-D trajectory management [6], flow management [7], distributed air traffic management [8] and advanced communication infrastructure [9]. The utility of simulation tools to perform design space exploration of metrics such as conflict ratio, airspace throughput, average flight time and average signal strength for small UAS traffic management was highlighted in [10].

The SIRIUS batch simulation framework incorporates established tools developed over the years for separation assurance and Detect and Avoid (DAA) research (e.g., ICAROUS). Many of the algorithms and models have been formally verified and/or extensively tested in simulation and flight tests. SIRIUS is designed to investigate the integration and interoperability of on-board DAA technologies with ground-based strategic separation services for both near-term (e.g., pilot in the cockpit, assistive automation), as well as future, advanced stages (e.g., unmanned vehicles, automated safety critical systems) for UAM operations.

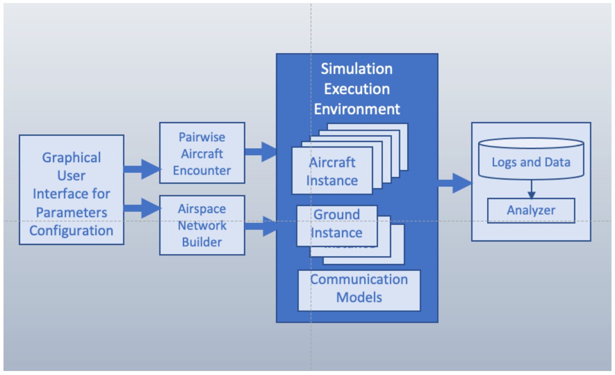

Fig. 1. SIRIUS Simulation Framework

At a high level, as shown in Fig. 1, SIRIUS provides a highly configurable Graphical User Interface to construct complex airspace interactions, different route structures containing waypoints, vertiports, air corridors, merge and crossing fixes, and keep-in/out geofences. Time of arrival constraints can be imposed at waypoints/fixes on a per vehicle basis. The Pairwise Aircraft Encounter and Airspace Network Builder allows the design of pairwise or structured encounters. The Simulation Execution Environment enables the creation of multiple simulated aircraft types, (Aircraft Instance in Fig. 1), that currently include small rotorcraft, lift+cruise, and fixed wing General Aviation (GA) vehicles. These aircraft can be configured as DAA equipped (e.g., ICAROUS, ACAS Xr) or non-equipped. SIRIUS provides a modular interface to ground-based services, (Ground Instance in Fig. 1), that enables the integration of different elements of a UAM ecosystem, such as a ground-based traffic surveillance system or a ground operator. For example, if a predicted conflict is detected during a simulated encounter, an avoidance maneuver can be autonomously computed and executed by the onboard decision-making logic, handled in coordination with a ground system, or left unresolved in case of non-cooperative vehicles. The simulation environment also supports modelling of wind, navigation and sensor uncertainties, and communication delays.

SIRIUS provides a data interchange format to record, replay, and analyze simulation output in an attempt to standardize simulation efforts across UAM stakeholders. An interactive graphical analysis capability helps isolate, visualize, and compare relevant vehicle state data and metrics. Standard and widely used measures of performance metrics for detect and avoid are computed and plotted.

III. SIRIUS SUB-SYSTEMS

A. ICAROUS

ICAROUS (Integrated Configurable Architecture for Reliable Operation of Unmanned Systems) [11] is an on-board software system of distributed services for autonomous UAS. Its core functions include monitoring, and controlling for potential violations to conformance criteria dynamically updated during flight, such as new or updated geofences, standoff distances, flight plan updates, and traffic surveillance data as well as resolving contingencies and supporting mission flight requirements. A central element of safety assurance for

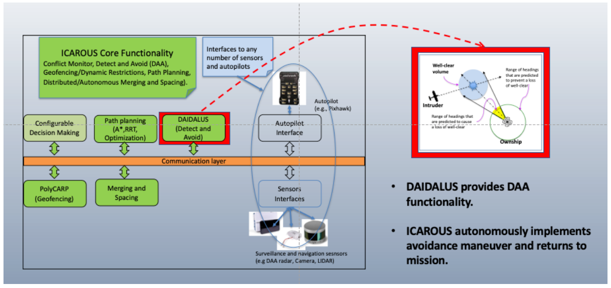

Fig. 2. ICAROUS High Level Architecture

autonomy is the ability to detect and identify possible conflicts with other users of the airspace as well as stationary obstacles on the vehicle path, avoiding collisions and maintaining a safe distance from all possible threats. In addition, ICAROUS actively computes resolution and recovery maneuvers and safe, conflict-free return to mission routes that can be configured by the user. Several of the algorithms in ICAROUS, such as, DAIDALUS (DAA) and PolyCARP (Geofencing), have been formally verified and/or extensively tested both in simulation or flight (e.g., Over 400 flight tests under the UTM project.)

ICAROUS path conformance monitoring and 4D trajectory compliance are fundamental capabilities for the highly constrained future UAM operations in which aircraft will me expected to fly very precise, time-constrained trajectories. More details on the individual functions depicted in Fig. 2 can be found in [11].

The C++ version of ICAROUS is available under NASA Open-Source Agreement (https://github.com/nasa/icarous).

B. DAIDALUS

ICAROUS detect and avoid capabilities are built upon DAIDALUS [12], a NASA-developed software library for Detect and Avoid included as the reference implementation for DAA by the RTCA DO 365 MOPS (Phase 1: En-route), DO 365A MOPS (Phase 2: Terminal Area -- DAIDALUS v2.0.1) and DO 365B MOPS (Phase 3: Non-cooperative aircraft -- DAIDALUS v2.0.2).

DAIDALUS uses a parametric volume, referred to as the well-clear volume (WCV), such that aircraft pairs jointly occupying this volume are considered to be in a well-clear violation. DAIDALUS includes algorithms for predicting a potential well-clear violation within a given look-ahead time, assuming non-maneuvering trajectories as well as for determining the instantaneous well-clear status between a pair of aircraft. Furthermore, DAIDALUS implements algorithms for computing maneuver guidance in the form of conflict bands, assuming a simple kinematic trajectory model for the ownship aircraft. Conflict bands represent ranges of track, ground speed, vertical speed, and altitude maneuvers that are predicted to result in well-clear violation with one or more traffic aircraft within a given look-ahead time. When aircraft are not well clear, or when a well-clear violation is unavoidable, DAIDALUS computes well-clear recovery bands, which represent ranges of horizontal and vertical maneuvers that regain well-clear status within the minimum possible time, while minimizing collision risk.

C. COMMUNICATION MODELS

To allow simulated aircraft to interact, SIRIUS uses a simple model of point-to-point radio communication. The model passes data between aircraft, inserts a configurable latency, and provides a probability that a given message will be successfully received and decoded by any other aircraft in the simulation. Alternative models can be selected depending on the environment being simulated. For example, some models include the fading effects expected in a dense urban area. The parameters of the model, and of the individual transmitters and receivers, can be selected to represent the expected performance for a given communication system.

The models can be used to broadcast position data to simulate ADS-B. Alternatively, the messages can contain coordination data to enable distributed, vehicle-to-vehicle merging and spacing. If there is a ground system in the simulation, it can send data to the aircraft, for example surveillance data from a ground-based radar model.

IV. ARCHITECTURE

The SIRIUS tool is implemented using a server-client architecture as shown in Fig 3. The client application provides a desktop graphical interface for user interaction while services such as aircraft simulation, data visualization and analysis are provided by one or more server applications. The various components of the SIRIUS architecture are described below.

A. Aircraft Instance

An Aircraft Instance comprises of a closed-loop aircraft dynamics model combined with core ICAROUS functionality such as trajectory guidance, traffic management, path planning and high-level decision making. The closed-loop aircraft dynamics model is highly configurable and modular. It takes as inputs a reference heading, ground speed and vertical speed along with the current wind speed and direction. The 3-dimensional position along with the actual heading, ground speed and vertical speed of the aircraft are provided as outputs. The 4-D trajectory guidance functionality in ICAROUS enables the aircraft dynamic model to follow a specific flight plan from takeoff to landing. Capabilities such as traffic management, path planning and decision making can be enabled/disabled depending on the level of autonomous capabilities expected from a given aircraft.

B. Ground Instance

A Ground Instance enables integration of ground components and their interaction with vehicles in the airspace. For example, ground-based radar plays a crucial role in providing surveillance data to aircraft in the airspace. Another key ground-based component of the UAM ecosystem is a pre-flight, strategic deconfliction capability that would provide 4D flight plan representations of UAM operations before flight initiation. Such functionality could also be represented by a "ground instance" in the SIRIUS architecture.

C. Simulation Execution Environment

The Simulation Execution Environment is a container for one or more aircraft or ground instances. The execution environment is responsible for advancing the simulation time and synchronizing all aircraft and ground instances. The execution environment also simulates the vehicle-to-vehicle (V2V) and vehicle-to-ground interaction using various communication models. The communication models simulate ADS-B type interactions between aircraft.

Multiple simulation execution environments can be instantiated locally or remotely. A simulation manager synchronizes all local and remote simulation environments and coordinates data exchange between them making it appear as a single large simulation execution environment.

The Execution Environment takes as input a given *scenario* and outputs a log of timestamped simulation traces containing all aircraft state data. A *scenario* is an abstract datatype defining the simulation timespan, a set of aircraft initial conditions and their corresponding flight plans, geofences and configuration parameters, route structures (merge/crossing fixes), wind profile (wind speed and direction) for the simulation timespan and communication model parameters.

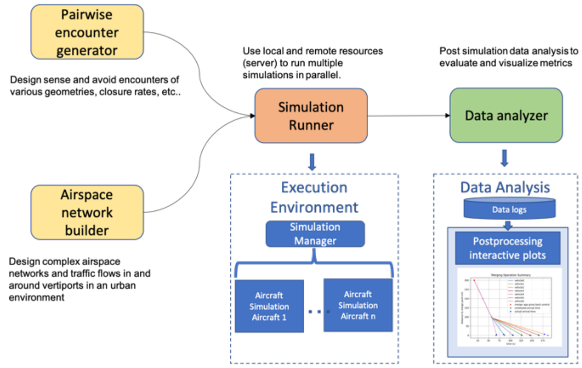

Fig. 3. SIRIUS workflow

D. *Scenario Generator*

The SIRIUS graphical user interface client, or desktop interface shown in Fig. 1, provides convenient options to design scenario configurations (e.g., initial conditions, flight plans, route structures, etc.), simulate and analyze the encounters with and without autonomous capabilities supplied by ICAROUS.

1) *Pairwise encounter generator*

The pairwise scenario generator is used to design specific DAA encounters between two aircraft. A pairwise encounter assumes that two vehicles are flying a constant ground track angle that may result in a well clear violation. The pairwise encounter is parameterized by the heading angles for the ownship and intruder, the horizontal and vertical speeds of each aircraft, well clear radius, time to well clear violation and a relative bearing angle. Alternately, a pairwise encounter can also be parameterized using the horizontal miss distance, time to closest point of approach and the relative track angle. The tool can generate an exhaustive batch of encounter geometries according to the ranges and discretization imposed on the encounter parameters or sample from the ranges according to a specific distribution (Monte Carlo sampling). This functionality is used extensively to investigate detailed performance of certain DAA parameter combinations, encounter geometries, winds, and other system parameters. Figure 4 illustrates a simple pairwise scenario setup.

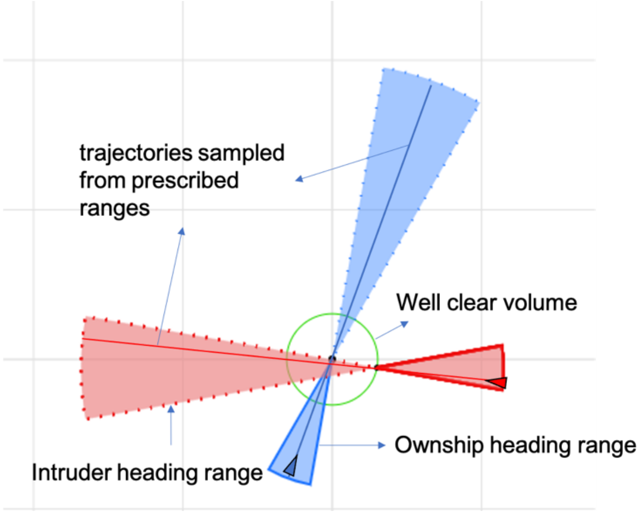

Fig. 4. Pairwise Encounter Generator

2) *Airspace Network Builder*

The network generator tool enables the construction of complex airspace route structures containing a network of vertiports, routes connecting various fixes including crossing and merging fixes. Furthermore, one can also specify the input and output flow rates for each vertiport. The input/output flow rate parameters in turn can be used to control the overall airspace density.

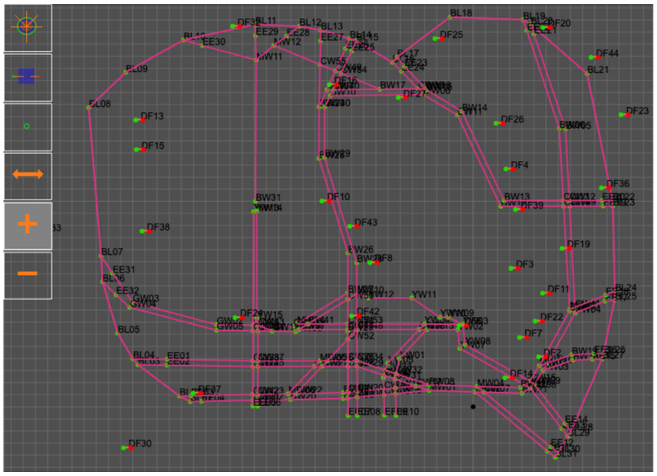

Fig. 5. An example UAM airspace containing a network of east-west and north-sound routes, vertiports and crossing/merge fixes.

The generator tools also provide options to configure the wind profile for the simulations and communication model parameters. Uncertainties associated with the ownship/traffic position and velocity can be specified by providing the noise covariances. Figure 5 illustrates an example airspace constructed with the Airspace Network Builder tool.

E. *Batch Simulation Runner*

The SIRIUS interface is agnostic to the underlying operating system (OS). To make SIRIUS usable on all operating systems, a method for connecting and sharing the compute resources of multiple machines over a local network was developed and built into the SIRIUS desktop interface. This gives the user the ability to either run small scenarios in parallel or to simulate a large scenario by efficiently managing available computing resources.

To create a cluster of locally networked machines, a lightweight server is started on each remote machine, then a web socket connection to the remote machine is created from the desktop interface. The remote servers spawn processes (i.e., aircraft simulation instances) as directed and reports status messages back to the desktop interface. If the desktop interface is run on a Unix based operation system, a local server process is automatically created, and a web socket connection is established. The local connection is treated the same as a remote connection to standardize messaging, process handling, and output file handling.

In SIRIUS, a batch is a set of either pairwise or network generated scenarios. When running a batch, the user is given the option to either process the scenario in parallel or in a distributed manner. Parallel processing runs multiple scenarios at the same time as separate processes across the cluster limiting the number of concurrent process to the number of available CPUs. Distributed processing splits the scenario based on the number aircraft being simulated in the scenario into separate processes. Data exchange between processes is enabled via an inter-process communication framework. Figure 6 illustrates the distributed simulation architecture.

SIRIUS automatically handles file movement to and from the remote servers to avoid unnecessary complication and errors created from file mismanagement. At the start of a batch, the batch, scenario, parameter, and flight plan files are all copied to each connection. Upon completion of a batch run the resulting log files are compressed on each machine and sent to the controller machine (host). The log files are then decompressed on the host and reorganized into the proper file structure for analysis.

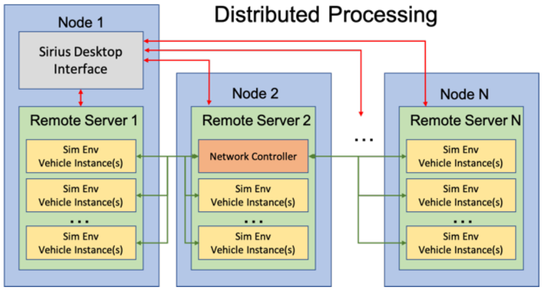

Fig. 6. Distributed simulation capability

F. Data Analyzer

SIRIUS includes desktop tools for analyzing the data logs produced during simulation. Each simulated aircraft produces a log of its own state history information, and aircraft equipped with ICAROUS generate additional logs of avoidance maneuvers and merging operations. The desktop interface allows users to plot data in multiple ways to analyze the evolution of aircraft encounters highlighting key events in the decision logic such as type and magnitude of avoidance maneuvers, timing of events, predicted and actual distances at the point of closest approach, etc. The data can also be plotted as a 2D or 3D animation replaying the events of the simulation. These tools are useful for reviewing an individual aircraft encounter to analyze the behavior and performance of the autonomous software.

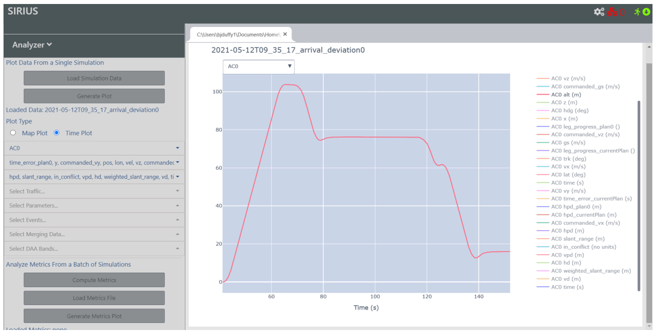

Fig. 7. SIRIUS Desktop Interface, showing altitude plot of simulated vehicle

The SIRIUS data analysis tools enable the study of aggregate measures of system performance as well as the examination of algorithm behavior during individual encounters. Currently, these metrics include large sets of established metrics of performance used for simulation and flight test data analysis. Data from multiple flights or simulations can be plotted together in histograms or scatter plots to review results from large batch simulations. SIRIUS outputs all metrics to comma separated text files that can be used for further statistical study. Figures 7 and 8 illustrate the data visualization capability in SIRIUS.

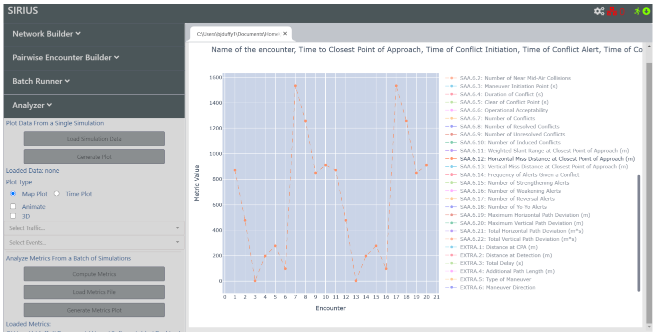

Fig. 8. SIRIUS Desktop Interface, showing a plot of the horizontal miss distance metric for a series of simulations

V. EXAMPLES

A. Simulations of Pairwise Detect and Avoid Encounters

A batch of 15 pairwise encounters is defined by uniformly sampling from the input parameters identified in Table 1. As mentioned in Section IV.D, these parameters define the geometry of an unmitigated encounter (i.e., no avoidance maneuver is executed). In this example, the predicted Horizontal Miss Distance (HMD) is sampled from a 100-110 ft range and the Time at Closest Point of Approach (TCPA) is sampled from a 20-21 second range. Figure 9 illustrates the trajectories resulting from simulating the 15 combined scenarios. The minimum horizontal distance between the vehicles during each scenario is presented in Figure 10.

Table 1: Pairwise Encounter Configuration

Parameters	**Values/Range**
HMD	100-110 ft
TCPA	20-21s
Relative Angle	230-280 degrees
Ownship speed	50-52 knots
Intruder speed	50-52 knots
Well clear radius	1000 ft (305m)

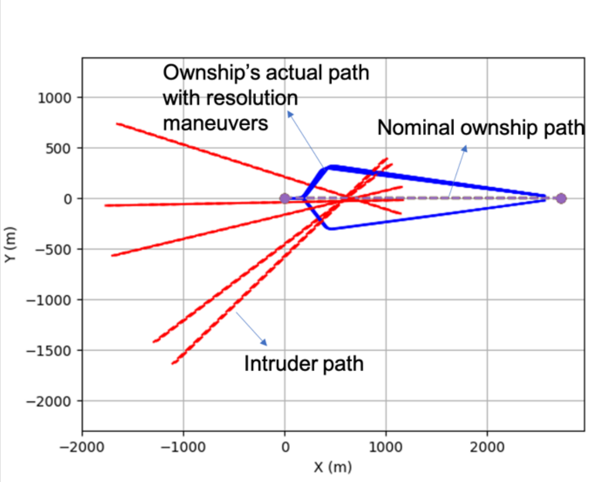

Fig. 9. Output trajectories from batch simulation of all scenarios

Note that the encounter geometries are defined such that the unmitigated response of the vehicles result in a horizontal separation of 100 ft at their closest point of approach. The mitigated responses observed from the batch simulations indicate that the ownship is able to maintain a horizontal separation of > 1000 ft (> 300 m) after avoidance maneuvers are executed

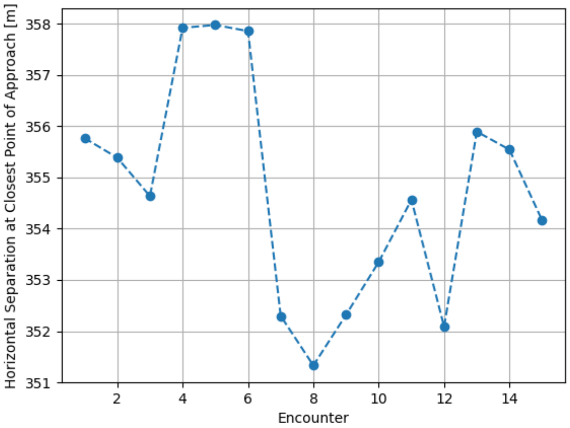

Fig. 10. Horizontal separation at closest point of appraoch

B. *Distributed Simulation of Large Scale Airspace Operations*

Figure 11 illustrates a simple route structure depicting UAM routes to/from vertiports. The route structure is defined in terms of a sequence of waypoints that include multiple merging and crossing fixes. Different route structures can be designed to explore the complexity and severity of traffic conflicts observed at various demand levels and wind conditions. Flights plans can be configured as a sequence of 3D way points with crossing time constraints and a required time of arrival at destination. Simulated aircraft equipped with onboard automation will maintain a required lateral navigation performance and autonomously make speed adjustments to satisfy required time constraints.

The scenario shown in Figure 11 illustrates a simulation snapshot of several concurrent trips between various vertiports. Such simulations can be used to analyze metrices related to airspace density, throughput of UAM route structures, evaluate risk related to loss of well clear or collisions, etc. For example, Figure 12 compares the final arrival time versus the scheduled time of arrival for each simulated trip in the scenario depicted in Figure 11. The vehicles are able to arrive at their destination as scheduled. However, incorporating winds and enroute conflicts (at crossing/merge fixes) can result in arrival delays. The difference between the scheduled vs actual time of arrivals can be used for establishing flight scheduling guidelines and also determine the required vertiport capacity (e.g., available takeoff/landing pads, etc.) to meet demand requirements at various hours of operation.

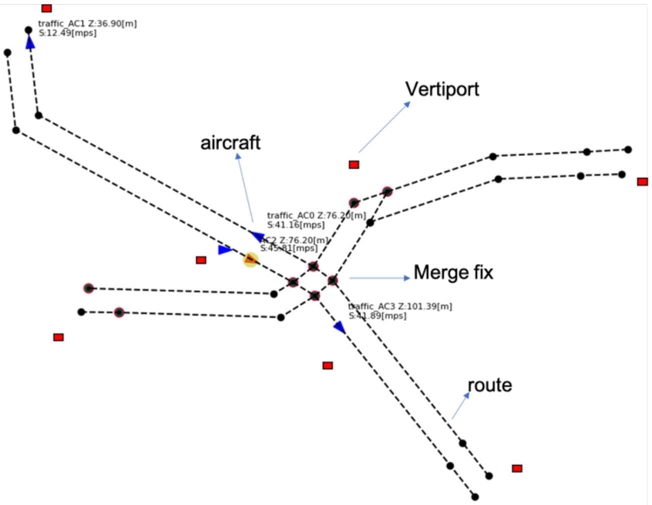

Fig. 11. Route structure containing airways connecting vertiports

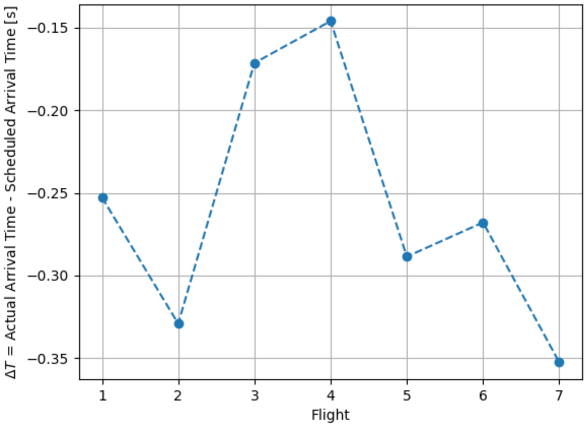

Fig. 12. Time of arrival analysis

The large-scale airspace simulations in SIRIUS provide an efficient way to study various performance-based separation concepts in which aircraft communication, navigation, and surveillance (CNS) models enable the evaluation of varying degrees of aircraft and system performance on separation requirements.

In the scenario in Figure 11, all traffic was modelled as ADS-B equipped, as described in Section III.C. Other onboard surveillance technology models (e.g., onboard radar, lidar, v2v, etc.) as well as ground-based surveillance sources (e.g., ground radar) can be incorporated into the simulations via the "surveillance data fusion" functionality which allows sensor uncertainty specification for each different source coupled with the ability to simulate various surveillance modalities.

Performance-based separation studies will need to address the impact of key CNS performance parameters on separation standards for UAM operations as well as existing users of the airspace and the air traffic system as a whole. Integration and interoperability studies will help inform future aircraft airworthiness standards as well as new airspace regulations and procedures.

Furthermore, the distributed airspace simulations can simulate aircraft with various physical performance constraints (e.g., turn rate, climb rate, acceleration constraints, etc.). Consequently, interactions between UAM other non-UAM aircraft (e.g., VFR traffic on final approach to a nearby runway) can be simulated to analyze the separation performance between different aircraft types

These studies need to be based on realistic traffic scenarios that incorporate current UAM industry trends and operations plans together with existing airspace users and representative equipage levels to explore the feasibility, safety, and potential efficiency gains of these new concepts.

VI. CONCLUSION

As the community converges to a new, shared understanding of the process by which UAM will become a reality in the short term, some major questions remain regarding scalability of future operations. Initial UAM operations, do present integration challenges that need to be addressed, but these operations are expected to be conducted under the current regulatory framework and airspace procedures. What levels of demand may require changes to the current ATM system and how the transition to a new paradigm that accommodates high demand levels will occur, are open questions.

Research efforts by NASA and the stakeholder community currently underway will continue exploring multiple concepts and technologies that could potentially enable scalable operations. Those concepts and technologies will continue to evolve as more knowledge is gathered.

The SIRIUS simulation framework is part of a suite of tools being developed by NASA to investigate the interoperability and safety performance of integrated air-ground separation and conflict management approaches that could enable scalable, safe UAM operations.

As a common understanding of UAM operational scenarios and representative traffic conditions is achieved, performance evaluation efforts will be formulated and conducted to support the development of separation standards and conflict management development.

Acknowledgements

The authors would like to acknowledge the work of the extended team of researchers whose work made possible the development of the SIRUS tool and to the UAM project for their support and encouragement. Special thanks to John Foster who developed the aircraft performance models for rotorcraft and eVTOL aircraft integrated into SIRIUS, Marco Feliu for his efforts on the integration of ACAS-Xr as an alternative CA method.

VII. REFERENCES

[1] NASA UAM Vision Concept of Operations https://ntrs.nasa.gov/api/citations/20205011091/downloads/UAM%20Vision%20Concept%20of%20Operations%20UML-4%20v1.0.pdf

[2] FAA NextGen Urban Air Mobility (UAM) Concept of Operations v1.0, 2020. https://nari.arc.nasa.gov/sites/default/files/attachments/UAM_ConOps_v1.0.pdf

[3] MITRE Urban Air Mobility Airspace Integration Concepts: Operational Concepts and Exploration Approaches, 2019 https://www.mitre.org/sites/default/files/publications/pr-19-00667-9-urban-air-mobility-airspace-integration.pdf

[4] Molina, M., Carrasco, S., & Martin, J. (2014, June). Agent-based modeling and simulation for the design of the future european air traffic management system: The experience of cassiopeia. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 22-33). Springer, Cham.

[5] Gurtner, G., Bongiorno, C., Ducci, M. and Miccichè, S., 2017. An empirically grounded agent based simulator for the air traffic management in the SESAR scenario. Journal of Air Transport Management, 59, pp.26-43.

[6] K. Margellos and J. Lygeros, "Toward 4-D Trajectory Management in Air Traffic Control: A Study Based on Monte Carlo Simulation and Reachability Analysis," in IEEE Transactions on Control Systems

Technology, vol. 21, no. 5, pp. 1820-1833, Sept. 2013, doi: 10.1109/TCST.2012.2220773.

[7] Wolfe, S. R., Jarvis, P. A., Enomoto, F. Y., Sierhuis, M., & van Putten, B. J. (2009). A multi-agent simulation of collaborative air traffic flow management. In Multi-Agent Systems for Traffic and Transportation Engineering (pp. 357-381). IGI Global.

[8] Peters, M., Ballin, M. and Sakosky, J., 2002, January. A multi-operator simulation for investigation of distributed air traffic management concepts. In AIAA Modeling and Simulation Technologies Conference and Exhibit (p. 4596).

[9] P. Park and C. Tomlin, "Investigating Communication Infrastructure of Next Generation Air Traffic Management," 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, 2012, pp. 35-44, doi: 10.1109/ICCPS.2012.12.

[10] Zhao, Z., Luo, C., Zhao, J., Qiu, Q., Gursoy, M.C., Caicedo, C. and Basti, F., 2019, April. A simulation framework for fast design space exploration of unmanned air system traffic management policies. In 2019 Integrated Communications, Navigation and Surveillance Conference (ICNS) (pp. 1-10). IEEE.

[11] María Consiglio, César Muñoz, George Hagen, Anthony Narkawicz, and Swee Balachandran, ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems, Proceedings of the 35th Digital Avionics Systems Conference (DASC 2016), Sacramento, California, 2016.

[12] César Muñoz, Anthony Narkawicz, George Hagen, Jason Upchurch, Aaron Dutle, María Consiglio, and James Chamberlain, DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems, Proceedings of the 34th Digital Avionics Systems Conference (DASC 2015), Prague, Czech Republic, 2015