Integration and Flight Test of Small UAS Detect and Avoid on A Miniaturized Avionics Platform

James G Lopez, Liling Ren, Baoluo Meng, Robert Fisher, Joel Markham
GE Research
Niskayuna, NY, USA

Michael Figard, Richard Evans, Ryan Spoelhof, Michael Rubenstahl, Scott Edwards
GE Aviation Systems
Grand Rapids, MI, USA

Igor Pedan
AiRXOS, Inc
Boston, MA, USA

Clark Barrett
Stanford University
Stanford, CA, USA

***Abstract*—Detect and avoid (DAA) all other aircraft is a critical component to enable small unmanned aircraft system (sUAS) beyond visual line of sight (BVLOS) operations. Derived from the version of Airborne Collision Avoidance System X (ACAS X) for large UAS (ACAS Xu), a new member of the ACAS X family for sUAS (ACAS sXu) is being developed by the Federal Aviation Administration's (FAA's) Traffic-Alert and Collision Avoidance System (TCAS) Program Office. ACAS sXu is intended to provide both collision avoidance (CA) and remain well clear (RWC) capabilities with both vertical and horizontal advisories for the remote pilot in command (RPIC) and/or automated response system onboard the aircraft. ACAS sXu is envisioned to utilize a standard logic to serve sUASs with different equipages and operating in different airspace domains. The standard ACAS sXu logic may be hosted either in the embedded environment on board the sUAS vehicle or in a Cloud environment such as a UAS traffic management (UTM) Service Suppler (USS) platform. It may be integrated with surveillance sources such as Automatic Dependent Surveillance-Broadcast (ADS-B), the anticipated remote identification (remote ID) tracking, networked/shared telemetry, airborne surveillance radar, and ground based surveillance radar, for both cooperative and non-cooperative intruders. To demonstrate proof of concept, gather surveillance data, verify simulation environment, and characterize early logic performance, the FAA and industry partners integrated DAA systems featuring the ACAS sXu logic Version 0, in both embedded environments and a Cloud environment, and successfully conducted a week-long flight test in October 2018 at the New York UAS Test Site in Rome, NY. This paper presents the integration of the sUAS DAA on a miniaturized avionics platform and flight test with a fixed-wing sUAS platform.**

Keywords—Airborne Collision Avoidance System X (ACAS X), Automatic Dependent Surveillance-Broadcast (ADS-B), avionics, detect and avoid (DAA), flight test, Small unmanned aircraft system (sUAS), surveillance.

I. Introduction

Detect and avoid (DAA) all other aircraft and yield the right-of-way to all aircraft, airborne vehicles, and launch and reentry vehicles is a critical component to enable small unmanned aircraft system (sUAS) beyond visual line of sight (BVLOS) operations [1]. Although several different DAA concepts have been developed in the past, such as the Air Force's Multiple Sensor Integrated Conflict Avoidance (MuSICA)/Jointly Optimal Conflict Avoidance (JOCA) [2] and the National Aeronautics and Space Administration's (NASA's) Detect and Avoid Alerting Logic for Unmanned Systems (DAIDALUS) [3], there is a gap in providing a standardized DAA capability for sUAS. Derived from the version of Airborne Collision Avoidance System X (ACAS X) for large UAS (ACAS Xu), the Federal Aviation Administration's (FAA's) Traffic-Alert and Collision Avoidance System (TCAS) Program Office is developing a new member of the ACAS X family for sUAS (ACAS sXu) to fill this gap.

To demonstrate proof of concept, gather surveillance data, verify simulation environment, and characterize early logic performance, the FAA and industry partners formed a team to integrate DAA systems featuring the ACAS sXu logic, in embedded environments onboard sUAS vehicle platforms and a ground-based Cloud environment in a UAS traffic management (UTM) Service Suppler (USS) platform, and successfully conducted a week-long flight test in October 2018 [4]. In addition to the FAA TCAS Program Office, the partner team included the Johns Hopkins University Applied Physics Laboratory (APL) (ACAS sXu), Massachusetts Institute of Technology (MIT) Lincoln Laboratory (ACAS sXu), the General Electric (GE) team (airborne/ground system integration, USS platform, avionics, and fixed-wing sUAS and flight operations), which included GE Global Research, GE Aviation Systems, and AiRXOS, Fortem Technologies (airborne system integration, airborne radar, and multirotor sUAS and flight operations), the Northeast UAS Airspace Integration Research Alliance (NUAIR) (test site and manned intruder aircraft), Griffiss International Airport (RME) in Rome, NY (test range), AX Enterprize (surveillance data), and Gryphon Sensors (ground radar).

The remainder of the paper reports the integration of ACAS sXu on a GE miniaturized avionics platform, system integration test, and the flight test with GE's fixed-wing sUAS, as outlined below. A companion paper [5] presents the assessment and the safety case of the ACAS sXu.

II. ACAS sXu Logic Development

ACAS sXu [5] like other ACAS X variants contains a set of logic and a series of numeric lookup tables that encode much of the decision making information. These tables are optimized offline, prior to being loaded onto the platform, in a compute intensive surrogate modeling process. This process uses a set of trajectory profiles across a number of airspaces along with an appropriately weighted objective function to compute the tables as part of a Markov Decision Process. For ACAS sXu version 0, the tables from the large Xu system were used without any re-tuning. However, range and speed indexing were independently scaled to account for the sUAS dynamics. In addition, the tables were downsampled such that the overall memory footprint was appropriate for the test platforms. This downsamping was performed in conjunction with simulation to ensure safety and operational suitability metrics were not compromised.

The tables are flown with accompanying DAA software, which is written in Julia. The complete Julia is documented as the sXu Algorithm Design Description (ADD) [7], and includes a parameter file that details the dozens of parameters which tune the real-time software portion of the system. In going from Xu to sXu, the active surveillance, transponder data and coordination entry points were removed as that functionality was not required (however future sXu versions will likely restore some of those functions as the concept and sUAS landscapes evolve). A ground surveillance report was added to facilitate the use of ground sensing networks as well as other pre-tracked 3D position data sources.

ACAS sXu includes features unique to its concept of use. Rather than including separate remain well-clear (RWC) and collision avoidance (CA) functions, sXu includes a single level of alerting and guidance, with the separation volume scaled based on intruder type. ACAS sXu includes real-time dynamic scaling, meaning that the separation volumes provided by the system can be adjusted in real-time based on system inputs and states. The ACAS sXu development team is working within the ASTM International's Committee F38 on Unmanned Aircraft Systems, RTCA Special Committees 147 and 228, and the UAS Science and Research Panel (SARP) to inform the scale factors, protection volumes and safety metrics of the final system.

III. Demonstration System

Figure 1 depicts the integrated system deployed and demonstrated during the October, 2018 flight tests. The test aircraft, an 84" fixed-wing Bushmaster from Legacy Aviation, was equipped with GE Aviation's low SWaP avionics computer. The ACAS sXu logic was integrated onto the avionics computer's software stack and existing air-ground datalink and ground control station (Section III-B).

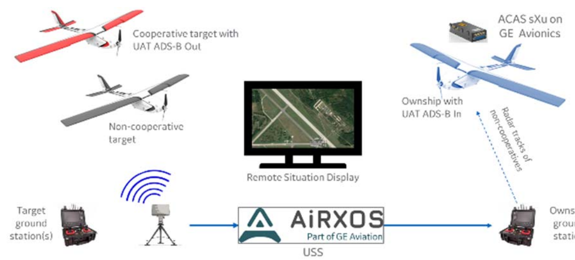

Fig. 1. Demonstration System

In total 33 sorties involving unmanned ownship and intruder and 4 sorties with unmanned ownship and manned intruder (Piper Cherokee) were flown with no safety violations. During the cooperative encounters, the ADS-B reports were fed into the ownship ADS-B receiver, and the reports were then processed by an ADS-B service that GE created and hosted on the M100 (Figure 2). The integrated ADS-B service formatted the ADS-B-In reports in a format that could be consumed by the onboard ACAS sXu logic. The ACAS sXu logic then generated advisories (Clear of Conflict. Weak Left, Hard Left, Weak Right or Hard Right). The ACAS sXu advisories were then transmitted to a ground control station (GCS) where the pilot in command (PIC) could choose to execute the advised collision avoidance maneuver. We also flew sorties where the intruder did not have an ADS-B transmitter. For the purposes of this test we referred to these types of intruders as non-cooperatives. The tracks of the "non-cooperatives" were generated by both ground based radar and telemetry data transmitted from the intruder to its ground control station. The "non-cooperative" intruder tracks were then fed to the AiRXOS USS, which were then sent to the ownship GCS for transmission to the ownship Bushmaster as inputs to the ownship's airborne ACAS sXu logic.

A. The M100 Miniaturized Avionics Platform

GE has developed a new family of open and scalable systems targeted at the next generation UAS platforms. One instantiation of this family is the M100 which provides an integrated complement of powerful computing resources to satisfy both vehicle and mission processing functions while minimizing SWaP and cost. This highly integrated system weighs just 24 ounces, is 6.5"×3"×1.5" and incorporates all functions necessary to not only fly and navigate a vehicle but also conduct a considerable amount of mission processing at the edge (i.e. onboard).

The M100 is built upon state-of-the-art System on Chip (SoC) technology, with the 'horsepower' to perform onboard sensor processing, software defined radio functions such as signals intelligence (SIGINT), datalink and communications, host autonomy applications, while performing real-time flight control and navigation functions. It includes an internal, 10 DOF navigation sensor package and a dual GPS receiver. The 10 DOF includes 3x Gyro, 3x Acceleration, 3x Magnetometer and Barometric Altimeter.

B. System & Software Architecture

Figure 2 depicts ACAS sXu and other necessary functions as secure containerized services hosted on the M100 along with the required integration with platform services.

The majority of mission and flight management services are

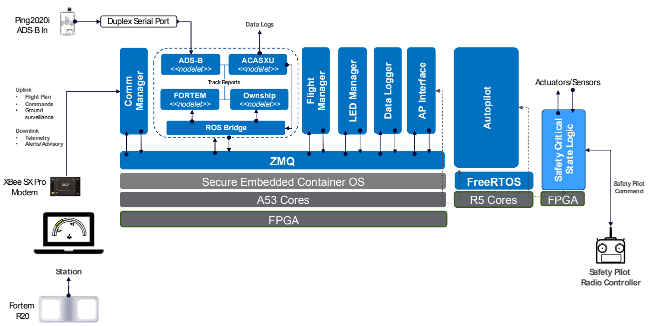

Fig. 2. System and Software Architecture

hosted on a secure embedded container operating system (OS). The OS is built with the Yocto project and contains all required services and device drivers to support onboard applications. The OS has been hardened from a security perspective and provides a small footprint option with limited attack surface.

The decision-making logic for the ACAS-sXu service execute on the OS. They are deployed as Linux containers which aids in portability to other embedded or enterprise (e.g. UTM) systems. Surveillance and collision avoidance services are implemented as a Robotic Operating System (ROS) nodes to allow easier integration with commercial and open-source based platforms. Support services for receiving track information from ADS-B and ground-based radar FORTEM, feed track reports to the ACAS-sXu service. ACAS sXu would then generate alerts based on the decision logic described in the previous section. The services for managing and sharing data with the rest of the platform services hosted on the airborne embedded system as well as ground-based systems (GCS) are implemented as Linux containers and use ZMQ as the messaging layer.

Lastly, an autopilot developed by GE Aviation Systems is hosted on FreeRTOS running on an R5 lockstep cluster which is also part of the SoM embedded system. The SoM also includes 2 FPGAs, one of which hosts a safety control logic for managing contingencies.

C. Sensor Integration

For the purposes of the October 2018 ACAS sXu flight tests we used uAvionix Ping 2020i ADS-B transceiver. The ownship and intruder aircraft were both equipped with a Ping 2020i. During the cooperative encounters, the ADS-B reports were fed into the ownship ADS-B receiver, and the reports were then processed by an ADS-B service that GE created and hosted on the M100 avionics architecture depicted in Figure 2. The integrated ADS-B service formatted the ADS-B-In reports in a format that could be consumed by the onboard ACAS sXu logic. The ACAS sXu logic then generated advisories (Clear of Conflict. Weak Left, Hard Left, Weak Right or Hard Right). The ACAS sXu advisories were then transmitted to a ground control station (GCS) where the pilot in command (PIC) could choose to execute the advised collision avoidance maneuver.

IV. Flight Test at the New York UAS Test Site

The October 2018 ACAS sXu flight tests were conducted at the New York UAS Test Site (NYUASTS) at Griffiss Airport in Rome, NY (RME).

The NYUASTS maintains COAs that allow large UAS to fly within the airspace volumes from more than 7,000 square miles of diverse airspace above varied terrain extending to Flight Level (FL) 250, and FL750 upon request. Additionally, sUAS can be flown anywhere within the RME Class D, SYR Class C, and Class G Airspace nationally up to 1200’ AGL within visual line of sight. The Test Site offers 1 SRC LSTAR radar, 1 Saab Sensis ASDX SMR, 1 Gryphon Sensors R1400 radar, all located at RME. Additionally, 1 R1400 located at NYS HLS Training Center 6 miles south of RME, 6 Saab Sensis Wide Area Multilateration (WAM) sensors scattered around the RME Class D. Command Center w/visualization tools that fuse all above sensor data into flight picture. 12,000 ft 200 ft runway, numerous hangars of varying size, expansive aprons (former B-52 base), Contract Control Tower at RME.

During the flight tests the ownship and intruder aircraft were tracked via one or more of the ground-based radars

V. Observations and Lessons Learned Flight Test

The list below summarizes the flights that were conducted some of the observations made during the testing

- 33 unmanned vs. unmanned sorties, no safety violations
- 4 unmanned vs. manned sorties, no safety violations
- Automatic and manual advisory response demonstrated
- Low air-to-air ADS-B reception probabilities (ADS-B was used as only a proxy to a future V2V link technology). Future versions of sXu will include a generic V2V entry point to enable the use of any number of solutions for cooperative communications appropriate for DAA.
- Air-to-ground ADS-B reception was near 100%. Recycling these receptions into airborne data recordings allows playback and analysis of the logic performance despite the poor air-to-air cooperative surveillance
- Encouraging non-cooperative surveillance effectiveness against manned and unmanned intruders.

VI. Future Directions

In future work, we plan to train a DNN (Deep Neural Network) on the look up table (LUT) associated with ACAS sXu. Stanford University has demonstrated that one technique for dramatically compressing the large numeric lookup table that constitutes the logic expression for ACAS Xu is to train a Deep Neural Network (DNN) to find a robust non-linear function approximation that represents the lookup table (LUT) [8]. It has also been shown that the DNN approximation of ACAS Xu performs better on certain safety metrics than does the LUT [8], and can significantly decrease the amount of storage. We believe that the DNN version of ACAS sXu will achieve similar improvements while reducing the amount of required storage. To further increase the trust-worthiness of DNNs, we will leverage the state-of-the-art satisfiability modulo theory [9] [10] [11] – Marabou [12] solver to perform robustness verification on the DNNs and form a formal proof that the network always behaves as intended, and will perform a flight-test again with the verified DNNs. However, much effort remains to demonstrate that ACAS sXu is safe and certifiable. This year GE Research in collaboration with Stanford University will train a DNN on the LUT associated with ACAS sXu. In addition, we will attempt to build a safety assurance case that could be used to prove several safety properties of the DNN manifestation of ACAS sXu. We will formally verify some properties of the ACAS sXu DNN using the Marabou tool being developed at Stanford. In order to generate more evidence of safety, especially for properties that cannot be formally proven, we will apply a robustness testing technique. Also, we will integrate and flight test the ACAS sXu DNN both in simulation and in actual flight tests with ACAS sXu integrated into GE's M100 avionics platform designed for small UAS with encounters to be flown on GE Research's small fixed wing test aircraft. We will apply an Adaptive Stress Testing (AST) technique to identify near mid-air collision (NMAC) encounters that stress the ACAS sXu logic. By flying these stressful NMAC encounters in simulation and in actual flight tests we will generate additional evidence of safe operation. All of the safety artifacts generated via the various means described will be assembled as evidence in a safety assurance case [13], [14] which will be constructed according to the rules of Goal Structuring Notation (GSN) [15]. A safety assurance case can be used to prove safe operation of safety critical software and is being considered as an alternative to the prescriptive DO-178 C process for generating and assembling safety evidence used to certify flight safety critical software.

VII. Summary

The ability for sUAS to detect and avoid all other aircraft is critical to enabling BVLOS operations and the large-scale integration of sUAS into the National Airspace (NAS). GE along with the FAA TCAS program office and other partners were the first to integrate and flight test the ACAS sXu version 0 logic, an important first step. GE plans to continue to work to integrate and flight test the ACAS sXu version 1, while working with Stanford University to implement the ACAS sXu v1 logic as a Deep Neural Network and to integrate and flight test the DNN logic as well as the original lookup table. The implementation of ACAS sXu version 1 results in a size reduction of 1000x and also offers the ability to apply new methods of formal verification and robustness testing using the Marabou tool under development at Stanford. We expect that the eventual certification of the ACAS sXu logic will benefit from the generation of evidence from a number of sources including simulation, flight testing, and formal verification of safety properties of the logic. We also expect that the generation of safety assurance cases using such techniques as Goal Structuring Notation (GSN) will be an important new tool for achieving affordable and scalable certification of UAS for BVLOS operation and operation in the NAS.

Acknowledgment

The authors thank Neal Suchy of the FAA's TCAS Program Office for his support and guidance. The authors thank Josh Silbermann of APL for his program leadership, Tyler Young, Benjamin Zintak, and Matt Lempka of APL for their collaboration and support, and Ian Jessen of MIT Lincoln Laboratory for building and providing the ACAS sXu libraries.

The GE Global Research (GRC) team thank Matt Vacanti and Jeff McGookey of GEAS for their tireless support and leadership throughout this effort, without them, none of this work would happen, Eric Armstrong of GEAS for his contribution in early flight tests, Ernie Thomas of GEAS for his help in airframe integration, Mike Dell'Anno of GRC for his guidance to the GRC team, Ben LaFergola and Yutao Zhou of GRC for their great effort in software integration.

The AiRXOS team thank Ray Chow of AiRXOS for his work in the weeks leading up and during the flight testing and Alex Baluha of AiRXOS for his development and operations support.

References

[1] FAA, "Waiver safety explanation guidelines for Part 107 waiver applications," guidance material webpage, including operational risks and mitigations questions, FAA, Washington, DC, December 21, 2018. Available at <https://www.faa.gov/uas/commercial_operators/part_107_waivers/waiver_safety_explanation_guidelines/> and <https://www.faa.gov/uas/commercial_operators/part_107_waivers/waiver_safety_explanation_guidelines/media/WSEG_operational_risks_mitigations.pdf> (accessed February 21, 2019.

[2] Graham, S., Chen, W., Luca, J. D., Kay, J., Deschenes, M., Weingarten, N., Raska, V., and Lee, X., "Multiple Intruder Autonomous Avoidance Flight Test," *Infotech@Aerospace 2011*, AIAA, Reston, VA, March 29-31, 2011.

[3] Davies, J. T. and Wu, M. G., "Comparative Analysis of ACAS-Xu and DAIDALUS Detect-and-Avoid Systems," NASA/TM-2018-219773, NASA Ames, Moffett Field, CA, February 2018.

[4] FAA, “ACAS sXu Brief-Out: Highlights and Preliminary Results from October 2018 Flight Test,” presented by B. Zintak, I. Jessen, L. Alvarez, J. Silbermann to the ASTM F38 DAA Performance Requirements Task Group (WK62668), TCAS Program Office, FAA, Washington, DC, December 10, 2018.

[5] Alvarez, L., Silbermann, J. M., “ACAS sXu: Robust Decentralized Collision Avoidance for sUAS,” to be published at *38th Digital Avionics Systems Conference*, IEEE, New York, NY, September 8-12, 2019.

[6] FAA, “Concept of Use for the Airborne Collision Avoidance System Xu for Small UAS (sXu),” Version 0, Revision 2, prepared by Bender, W., TCAS Program Office, FAA, Washington, DC, November 30, 2018.

[7] FAA, “Algorithm Design Description of the sXu Airborne Collision Avoidance System X,” working-draft, Version 0, Revision 2, prepared by Jessen I. and Klaus, R., TCAS Program Office, FAA, Washington, DC, June 01, 2018. [2]

[8] Julian, Kyle D., Mykel J. Kochenderfer, and Michael P. Owen. "Deep neural network compression for aircraft collision avoidance systems." *Journal of Guidance, Control, and Dynamics* 42.3 (2018): 598-608.

[9] Barrett, Clark, and Cesare Tinelli. "Satisfiability modulo theories." *Handbook of Model Checking*. Springer, Cham, 2018. 305-343.

[10] Meng, Baoluo, et al. "Relational constraint solving in SMT." *International Conference on Automated Deduction*. Springer, Cham, 2017.

[11] Meng, Baoluo. "*Satisfiability modulo relations: theory and applications.*" PhD (Doctor of Philosophy) thesis, University of Iowa, 2018.

[12] Guy Katz, Derek A. Huang, Duligur Ibeling et.al. The Marabou Framework for Verification and Analysis of Deep Neural Networks. In Proceedings of the 30th International Conference on Computer Aided Verification 2019, New York, NY

[13] E. Denney, G. Pai and I. Habli, "Towards Measurement of Confidence in Safety Cases," Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement, 2011.

[14] P. Graydon, J. Knight and E. Strunk, "Assurance based development of critical systems," 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07), 2007.

[15] T. Kelly and R. Weaver, "The goal structuring notation–a safety argument notation," Proceedings of the dependable systems and networks 2004 Workshop on Assurance Cases, 2004.