ICAROUS

INTEGRATED CONFIGURABLE ALGORITHMS FOR RELIABLE OPERATIONS OF UNMANNED SYSTEMS

María Consiglio, César Muñoz, George Hagen,
Anthony Narkawicz
NASA Langley Research Center
Hampton, VA
USA

Swee Balachandran
National Institute of Aerospace
Hampton, VA
USA

***Abstract*— NASA's Unmanned Aerial System (UAS) Traffic Management (UTM) project aims at enabling near-term, safe operations of small UAS vehicles in uncontrolled airspace, i.e., Class G airspace. A far-term goal of UTM research and development is to accommodate the expected rise in small UAS traffic density throughout the National Airspace System (NAS) at low altitudes for beyond visual line-of-sight operations. This paper describes a new capability referred to as ICAROUS (Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems), which is being developed under the UTM project. ICAROUS is a software architecture comprised of highly assured algorithms for building safety-centric, autonomous, unmanned aircraft applications. Central to the development of the ICAROUS algorithms is the use of well-established formal methods to guarantee higher levels of safety assurance by monitoring and bounding the behavior of autonomous systems. The core autonomy-enabling capabilities in ICAROUS include constraint conformance monitoring and contingency control functions. ICAROUS also provides a highly configurable user interface that enables the modular integration of mission-specific software components.**

Keywords—UAS; Detect and Avoid; Autonomy; UTM;

I. INTRODUCTION

NASA's Unmanned Aerial System (UAS) Traffic Management (UTM) project is a research and development effort aimed at enabling increasing number of low-altitude, small UAS operations in uncontrolled airspace. The UTM project addresses multiple aspects of airspace operations and air traffic management. In particular, UTM research concerns the technologies and capabilities necessary to enable safe, routine, small-UAS (sUAS) operations for civilian and public applications such as delivery of goods, infrastructure monitoring, precision agriculture, and search and rescue. One key objective of the UTM project is ensuring that the safety of existing users of the airspace is preserved. To ensure the desired target level of safety, while at the same time enabling efficient, extended, and increasingly more autonomous operations, small UAS will have to be equipped with a minimum set of core capabilities to satisfy the mission goals as well as safety and UTM constraints.

The UTM Concept of Operations [1] envisages UAS that will use "onboard detect and avoid systems to avoid other traffic, adverse weather, terrain, and man-made and natural obstacles." UTM incremental development process comprises a series of builds of increasing complexity and user services. In the initial stages, UTM ensures the safety of UAS operations with a combination of limited airborne technology, procedural segregation from other potential users of the airspace, and limited user services. Later builds assume increasing levels of operational autonomy assisted by airborne technologies of increasing complexity and reliability. The Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems software architecture (ICAROUS), being developed as part of the UTM project, will provide highly assured core software modules for building safety-centric autonomous unmanned aircraft applications. ICAROUS follows a long history of implementations of formally verified algorithms for safety-critical air traffic applications [2, 3, 4, 5, 6]. These implementations continue to evolve with new modules and algorithms to address newer challenges and problems in air traffic systems. A recent implementation known as DAIDALUS (Detect and Avoid Alerting Logic for Unmanned Systems) [7], which is the basis for the ICAROUS software architecture, is included as the reference implementation of RTCA-228 MOPS (Appendix G) for UAS DAA (Detect and Avoid). DAIDALUS consists of self-separation and alerting algorithms that provide situational awareness to UAS remote pilots in the form of maneuver guidance intended to aid in maintaining or regaining "well clear" separation for large unmanned vehicles. The safety-critical software components of ICAROUS follow a well-established formal development process. These include the formal specification and verification of the functional requirements and algorithms using the Prototype Verification System (PVS) [8] and rigorous software validation using methods such as model animation [9]. The design and development of highly assured software for safety critical applications is made possible by this rigorous process.

ICAROUS provides a mission agnostic and highly configurable API (Application Program Interface) to enable its integration to diverse mission applications and platforms. The software development will be made available under NASA's Open Source Agreement.

The remainder of this paper is organized as follows. Section 2 provides a brief description of the concept of use and high-level system architecture for ICAROUS-enabled autonomous operations in the UTM system. Section 3 introduces ICAROUS architecture and main functionality, Section 4 describes the safety-focused formal algorithm design and development approach used in ICAROUS, and Section 5 provides a summary of ongoing and future work.

II. Autonomous Operations in the UTM System

The UTM system aims at providing operation approval, geofences/constraints, and runtime flight monitoring to participating small, low altitude UAS operating in uncontrolled airspace. These vehicles, referred to as UTM Clients, are expected to communicate with UTM service providers, before and during operations, to ensure that the safety of the airspace and the non-flying public is preserved.

Within the UTM framework, vehicles equipped with autonomy-enabling technology will be capable of executing extended, beyond line-of-sight missions in increasingly complex, dense airspace conditions, using “onboard detect and avoid systems to avoid other traffic, adverse weather, terrain, and man-made and natural obstacles” [1]. Autonomous UTM clients will also be expected to submit flight plans for approval, as well as request geofencing information and airspace constraints from the UTM server, both before and during flight. The onboard systems will be responsible for the safe execution of the mission ensuring that airspace constraints are satisfied at all times. The onboard systems will be responsible for ensuring mission-safety by preventing violations of geofencing constraints and preserving the safety of other users of the airspace as well as the non-flying traffic. Fig. 1 depicts a multi-layer safety hierarchy designed to support Autonomous UAS Operations.

The safety hierarchy comprises multiple layers of complementary functionality designed to enable the integration of UAS missions of increasing autonomy for extended operations in non-segregated airspace.

The ground-based UTM systems provide the strategic airspace management functions to approve and establish airspace boundaries for the requested missions. On-board systems ensure compliance with the given airspace constraints and autonomously managing contingencies such as conflicts with other users of the airspace or stationary obstacles on the path of the flight.

The onboard system shown in Fig. 1 includes two complementary capabilities: ICAROUS and SAFEGUARD [10]. While both modules are capable of operating stand-alone, this implementation proposes an integrated architecture in which ICAROUS performs dynamic constraint monitoring, Detect and Avoid (DAA), obstacle avoidance and autonomous mission support, communicating with the ground systems and onboard sensors. SAFEGUARD independently monitors conformance to airspace constraints, utilizing highly-assured data and criteria established prior to flight. During flight, SAFEGUARD provides proximity boundary warnings to ICAROUS and autonomously terminates flight when constraint criteria are about to be violated or if the vehicle is unresponsive. Both ICAROUS and SAFEGUARD perform complementary safety functions and both systems are based on highly-assured, formally verified algorithms.

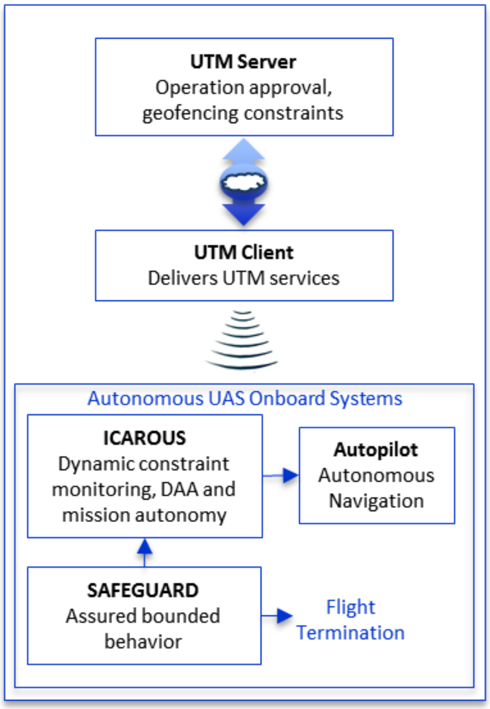


Fig. 1. Safety Layers for Autonomous UAS Operations

III. ICAROUS Functional architecture

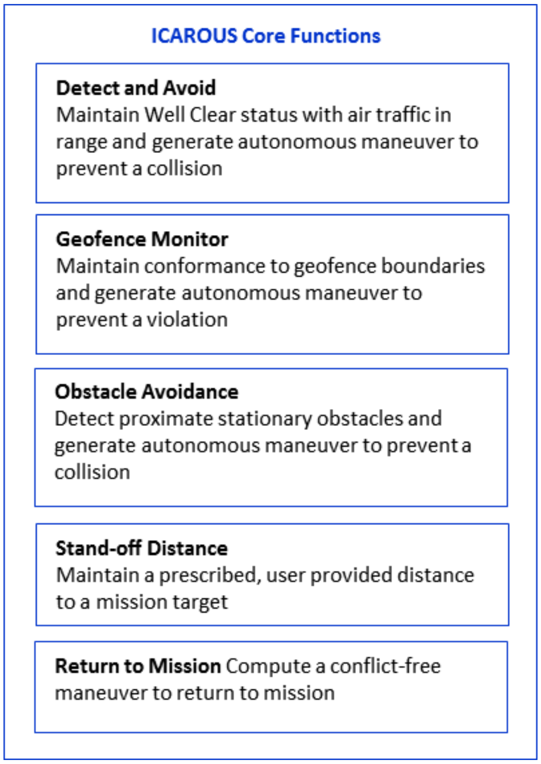
ICAROUS supports mission execution with autonomous, safety-based navigation and decision making while monitoring UTM procedures and mission-specific criteria. Its core functions, shown in Fig. 2, include monitoring, and controlling for potential violations to conformance criteria dynamically updated during flight, such as new or updated geofences, stand-off 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 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. The following sections provide a brief description of ICAROUS core functionality.

Detect and Avoid

ICAROUS detect and avoid capabilities are built upon DAIDALUS, a NASA-developed software library that implements a DAA concept for the integration of UAS in the National Airspace System (NAS). 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 of 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.


ICAROUS Core Functions

Detect and Avoid Maintain Well Clear status with air traffic in range and generate autonomous maneuver to prevent a collision

Geofence Monitor Maintain conformance to geofence boundaries and generate autonomous maneuver to prevent a violation

Obstacle Avoidance Detect proximate stationary obstacles and generate autonomous maneuver to prevent a collision

Stand-off Distance Maintain a prescribed, user provided distance to a mission target

Return to Mission Compute a conflict-free maneuver to return to mission

Fig. 2. ICAROUS Core Functions

Finally, DAIDALUS implements an alerting algorithm that computes an alert level indicating the severity of a potential well-clear violation as indicated in the RTCA SC228 Minimum Operational Performance Standards (MOPS), currently under development[1]. All these algorithms have been formally verified to be correct with respect to their functional requirements. The software implementations in Java and C++ have been rigorously validated against their formal models using a battery of stressing case scenarios.

DAIDALUS algorithms will be extended to support autonomous operations of sUAS and adapted to the airspace and traffic conditions appropriate for these missions. First, the well-clear volume used by the detection algorithm will be configured with threshold values appropriate for small UAS vehicles flying at low altitudes and speeds. Similarly, the maneuver guidance algorithms will be extended to support the flight dynamics of small rotorcraft and fixed-wing vehicles. In contrast to DAIDALUS, which assumes that a remote pilot will implement maneuver guidance to maintain or regain well-clear status, DAA maneuvers will be autonomously executed by ICAROUS. Therefore, ICAROUS includes a software module that selects a particular maneuver according to a user-defined optimization goal and fitness function. This optimized maneuver is checked against the mission-specific conformance criteria.

A. *Geofence Conformance*

Narkawicz and Hagen describe the mathematical foundations of the functions used to detect potential infringements of geofence boundaries in the constraint conformance algorithm [11]. These functions detect collisions between a linearly moving point, e.g. aircraft, and a (possibly moving) simple polyhedron generated by a 2 dimensional polygon and a pair of maximum/minimum altitudes. The collision detection algorithms have been formally verified in PVS to be mathematically correct under the assumption of bounded computer arithmetic errors. An implementation of these algorithms is used in a weather avoidance software module of the Stratway system [12]. This implementation will be integrated into ICAROUS for monitoring conformance to exclusionary (stay-off) or inclusionary (stay-in) geofence boundaries.

B. *Obstacle Avoidance*

The current instantiation of ICAROUS assumes that the mission area is clear of unexpected obstacles such as trees, birds, people, and objects. However, the software architecture enables the extension of ICAROUS with obstacle sensing and avoidance algorithms, which will be made available in future instantiations of the system. A basic 2-D obstacle detection algorithm that uses the OpenCV open source library (http://opencv.org) is currently under development.

C. *Stand-off Distance*

ICAROUS stand-off distance function illustrated in Fig. 3, monitors and maintains a pre-established distance to a inspection mission target such as a power line.

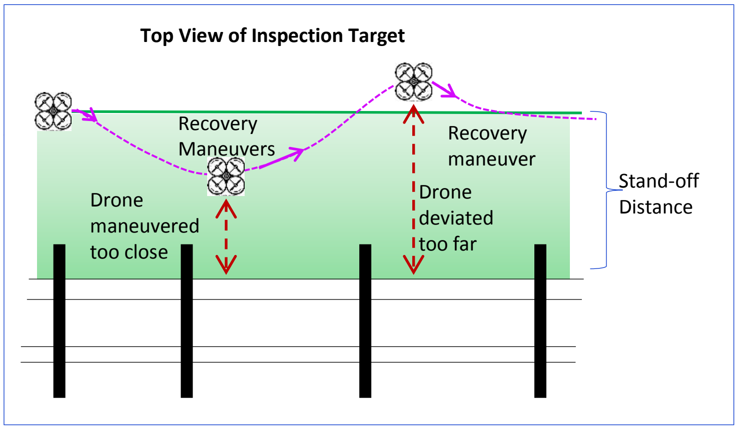


Fig. 3. Stand-off Distance Function

[1] DAIDALUS is the DAA reference implementation included in SC228 DAA MOPS, Appendix G.

If the vehicle deviates from the prescribed distance, ICAROUS generates a corrective maneuver to reestablish the appropriate stand-off distance. This functionality takes into account the dynamics and performance limits of the vehicle and other conformance criteria. The stand-off distance can be dynamically updated during flight, based on the needs and requirements of the mission.

The ICAROUS software architecture is organized into several modules to facilitate its applicability across a wide range of missions. A data acquisition module accepts input data from multiple onboard navigation and sensor sources. A communication module implements all interaction with other onboard applications as well as ground control. A hierarchically organized, deterministic finite state machine in ICAROUS handles the core decision making and planning functionalities, monitors mission progress and triggers appropriate functions according to a their established priority in the hierarchy. Data transfer between ICAROUS, the onboard autopilot and other mission specific applications take place via a standardized communication protocol (MAVLink[1]).

IV. Formally Verified Algorithms in Autonomous Systems

Key to the ICAROUS design is the use of highly assured algorithms that support the development of safety-centric, autonomous UAS applications. These highly assured algorithms follow a formal development process that includes:

- Formal specification of operational and functional requirements.
- Formal verification that the functional requirements satisfy operational requirements.
- Formal specification of functional models, i.e., algorithms.
- Formal verification that functional models satisfy functional requirements.
- Implementation of functional models as software modules.
- Rigorous validation that software implementations and functional models coincide in a set of stressing cases.

The PVS tool supports this formal development process. PVS (https://csl.pvs.sri.com) is a verification system developed by SRI International. It consists of a specification language to formally describe mathematical objects, such as algorithms and their properties, and a powerful interactive theorem prover. The theorem prover enables the mechanical verification of mathematical statements, such as the mathematical correctness of algorithms with respect to their specifications. The mathematical correctness property of an algorithm states that under well-specified assumption on the inputs of the algorithm, the algorithm always terminates in a state that satisfies a well-specified assumption on the inputs and outputs of the algorithm. In particular, a formally verified algorithm in PVS always converges to a solution. These formal proofs provide a high level of assurance that the formal models of the algorithms and their software implementations are correct.

A major challenge in the software verification of autonomous systems such as ICAROUS is that they frequently rely on highly complex, non-deterministic software modules that are not amenable to current verification techniques. ICAROUS design consists of a suite of highly assured, formally verified modules but it accommodates the incorporation of mission specific, user defined optimization techniques that may be required for certain applications. The notional graph in Fig. 4 shows the fundamental stages in the logic underlying the ICAROUS functions. The first stage in the process consists of a formally verified "Detector" function that identifies potentially unsafe conditions such as proximity to a geofence boundary or a predicted "well clear" violation with another aircraft. The second stage, the "Resolver", generates ranges of safe maneuvers that, based on the performance limitations of the vehicle, can be executed to avoid the detected potentially unsafe condition. Both the detector and the resolver algorithms are formally verified and proven correct as shown in [13]. However, selection of the optimal maneuver from the provided maneuver ranges depends on the application or specific use of the software. The selection may be based on a mission-specific goal, computed with a sophisticated optimizer which cannot be formally verified. The last stage in the process, the Correctness Criteria function, ensures the output conforms to the established criteria. Furthermore, the formally verified resolver guarantees that a non-optimized, but correct, maneuver always exists, in case the optimizer fails to converge to a solution.

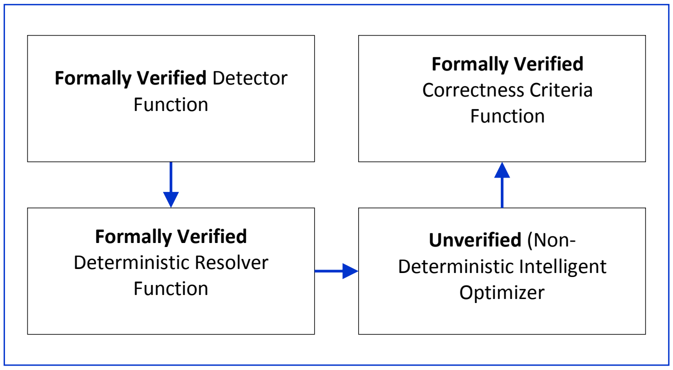

Formally Verified Detector Function
Formally Verified Correctness Criteria Function
Formally Verified Deterministic Resolver Function
Unverified (Non-Deterministic Intelligent Optimizer

Fig.4. Formally Verified Algorithms for Assured Safety

V. Conclusion

This paper presents a high-level description of the ICAROUS (Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems) software architecture, which is being developed under NASA's Unmanned Aerial System (UAS) Traffic Management (UTM) project. ICAROUS supports the modular development of safety-centric applications for small UAV autonomous missions. Key to the ICAROUS design is the use of highly assured algorithms that enable mission execution with autonomous, safety-based navigation and decision making while monitoring UTM procedures and mission-specific criteria. The set of ICAROUS algorithms includes detect and avoid, geofence conformance, stand-off distance monitoring, return to mission, and obstacle detection and avoidance. The rigorous formal development process used in ICAROUS and its highly configurable

[1] MAVLink is a lightweight, message marshalling library for UAS applications. It is widely used by off-the-shelf autopilots such as PX4, Pixhawk, APM, and Parrot AR Drone platforms.

architecture enable the integration of unverified, mission specific software components. The current instantiation of ICAROUS is being developed to support beyond visual-line-of-sight autonomous power lines inspection missions.

ACKNOWLEDGMENT

Special thanks to Connor Brooks and Alex Rivera for their contribution to the visual obstacle detection logic and their support to the ICAROUS project. The authors are also very grateful to Evan Dill and his summer students (Kyle Edgerton, Russell Gilabert, Nathan Lowe, and Serena Pan) for their enthusiastic support of the ICAROUS flight demonstration.

REFERENCES

[1] T. Prevot, J. Rios, P. Kopardekar, J. Robinson III, M. Johnson, and J. Jung, "UAS Traffic Management (UTM) Concept of operations to safely enable low altitude flight operations," 16th AIAA Aviation Technology, Integration, and Operations Conference, AIAA Aviation, (AIAA 2016-3292).

[2] G. Dowek, A. Geser, and C. Muñoz, “Tactical conflict detection and resolution in a 3-D airspace,” Proceedings of the Fourth International Air Traffic Management R&D Seminar ATM 2001, 2001.

[3] J. Maddalon, R. Butler, A. Geser, and C. Muñoz, “Formal verification of a conflict resolution and recovery algorithm,” Technical Paper, NASA/TP-2004-213015, April 2004.

[4] A. Galdino, C. Muñoz, and M. Ayala, “Formal verification of an optimal air traffic conflict resolution and recovery algorithm,” Lecture Notes in Computer Science, Vol. 4576, pages 177-188, 2007.

[5] A. Narkawicz, C. Muñoz, and G. Dowek,” Provably correct conflict prevention bands algorithms,” Science of Computer Programming, Volume 77, Issues 10-11, pages 1039-1057, 2012.

[6] A. Narkawicz, C. Muñoz, and G. Hagen,” An independent and coordinated criterion for kinematic aircraft maneuvers,” Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA-2014-2859, Atlanta, Georgia, 2014.

[7] C. Muñoz, A. Narkawicz, G. Hagen, J. Upchurch, A. Dutle, M. Consiglio, and J. Chamberlain,” DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems, Proceedings of the 34th Digital Avionics Systems Conference (DASC 2015), Prague, Czech Republic, 2015.

[8] S. Owre, J. Rushby, and N. Shankar, “PVS: A prototype verification system,” Lecture Notes in Artificial Intelligence, Vol. 607, pages 748-752, 1992.

[9] A. Dutle, C. Muñoz, A. Narkawicz, and R. Butler,” Software validation via model animation,” Lecture Notes in Computer Science, Vol. 9154, pages 92-108, 2015.

[10] E. Dill, S. Young, and K. Hayhurst, SAFEGUARD: An Assured Safety Net Technology for UAS,” (Unpublished) 2016

[11] A. Narkawicz and G. Hagen, “Algorithms for collision detection between a point and a moving polygon, with applications to aircraft weather avoidance,” 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA-2014-2859, Atlanta, Georgia, 2014.

[12] G. Hagen, R. Butler, and J. Maddalon, “Stratway: A Modular approach to strategic conflict resolution,” 11th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), number AIAA-2011-6892, 2011

[13] A. Narkawicz, C. Muñoz, “Formal verification of conflict detection algorithms for arbitrary trajectories,” Reliable Computing, Vol. 17(2), 2012.