UAS Autonomous Hazard Mitigation through Assured Compliance with Conformance Criteria

Evan T. Dill[1], Kelly J. Hayhurst[2], Steven D. Young[3] and Anthony J. Narkawicz[4]
NASA Langley Research Center, Hampton, VA, 23665

The behavior of a drone depends on the integrity of the data it uses and the reliability of the avionics systems that process that data to affect the operation of the aircraft. Commercial unmanned aircraft systems frequently rely on commercial-off-the-shelf and open source avionics components and data sources whose reliability and integrity are not easily assured. To mitigate failure events for aircraft that do not comply with conventional aviation safety standards, operational limitations are typically prescribed by regulators. Part 107 of the Federal Aviation Regulations serves as a good example of operational limitations that mitigate risk for small unmanned aircraft systems. These limitations, however, restrict growth possibilities for the industry. Any reasonable path toward achieving routine operation of all types of drones will have to address the need for assurance of avionics systems, especially their software. This paper discusses the possibility of strategically using assured systems as a stepping stone to routine operation of drones. A specimen system for assured geofencing, called Safeguard, is described as an example of such a stepping stone.

I. Nomenclature

ASTM	=	American Society for Testing and Materials
COTS	=	Commercial-Off-The-Shelf
DPAL	=	Data Processing Assurance Level
DAL	=	Design Assurance Level
DME	=	Distance Measuring Equipment
EASA	=	European Aviation Safety Agency
FAA	=	Federal Aviation Administration
FARs	=	Federal Aviation Regulations
GNSS	=	Global Navigation Satellite Systems
GPS	=	Global Positioning System
IMU	=	Inertial Measurement Unit
INS	=	Inertial Navigation System
JARUS	=	Joint Authorities for Rulemaking of Unmanned Systems
PNT	=	Position, Navigation, and Timekeeping
TACAN	=	TACtical Air Navigation
UAS	=	Unmanned Aircraft System
UTM	=	UAS Traffic Management
VOR	=	Very high frequency Omnidirectional Range
V&V	=	Verification and Validation

II. Introduction

ALMOST twenty years ago, David Hughes wrote, “Information technology is becoming a key part of everything the aerospace and defense industry does for a living, and as the century closes it is computers and software that hold the keys to the future.” [1] Advances in avionics technology undoubtedly have played a key role in the rapid

[1] Aerospace Research Engineer, Safety-Critical Avionics System Branch, Member.
[2] Senior Aerospace Research Engineer, Safety-Critical Avionics System Branch.
[3] Senior Aerospace Research Engineer, Safety-Critical Avionics System Branch, Associate Fellow.
[4] Aerospace Research Engineer, Safety-Critical Avionics System Branch.

growth of the commercial unmanned aircraft systems (UAS) or drone industry. Growth has been facilitated further by the ability to leverage open source and commercial-off-the-shelf (COTS) software and hardware (i.e., non-aviation grade avionics) to produce capable drones at a very low price point. Use of non-aviation grade systems, however, introduces uncertainty in terms of unexpected and undesirable events that are the nemesis of safety risk management in commercial aviation. Formulating a reasonable path toward routine operation of drones that do not meet conventional aviation system performance standards has been a challenge.

In 2016, the Federal Aviation Administration (FAA) released Part 107 of the Federal Aviation Regulations (FARs) prescribing rules for commercial use of small UAS [2]. Per Part 107, small UAS operations must comply with a number of operational limitations including prohibitions to flying beyond visual line-of-sight, flying within five miles of a commercial airport, or flying beyond 400 feet above ground level without an authorized exception. Other countries are proposing similar requirements for small UAS [3]. The operational limitations reduce risk by lowering the possibility of harm to people and property on the ground and to other aircraft in the airspace largely by controlling the environment in which the drone can operate. These limits, however, constrain the number and types of possible drone operations. Easing the regulatory restrictions to allow a more extensive range of drone types and operations, while maintaining safety, is essential to continued industry growth. The safety of such operations will inevitably depend on the reliability of the avionics systems.

The acceptable level of safety for manned (conventionally piloted) aircraft is generally very high because human life is always at risk. As such, avionics systems in most civil aircraft must comply with recognized aviation safety standards [4-7]. The cost of compliance can be significant [8] and has motivated efforts to streamline certification processes [9]. For unmanned (remotely-piloted and autonomous) aircraft, there is a much greater range in safety risk, and that risk depends inescapably on the operational environment [3, 10]. For example, safety risk associated with the loss of control of a drone in a remote, uninhabited environment may be very different from safety risk for the same event in an urban environment. Therein lies a thread of distinction for unmanned operations that could influence their assurance requirements and their attendant costs. The distinction may allow innovative approaches to mitigate safety risk associated with avionics systems failures.

One possible approach is to strategically use aviation-grade components to provide an assured bound on the behavior of systems that lack such a pedigree [11]. This approach utilizes the simplex architecture concept: using a very simple system that is cost-effective to certify to constrain the behavior of a more complex system that is more difficult and costly to certify [12]. Using an assured geofence to bound the navigation behavior of a COTS autopilot is an example of that approach. Geofencing in this context refers to constraining a UAS to an airspace defined by geographical and altitude limits and keeping a UAS from entering prohibited airspace. Geofencing capability has been recognized as fundamental for mitigating safety risk in UAS traffic management (UTM) [13], proposed standards from the American Society for Testing and Materials (ASTM) [14], guidelines from the Joint Authorities for Rulemaking of Unmanned Systems (JARUS) [15], and in the European Aviation Safety Agency's (EASA) UAS framework [16, 17].

This paper discusses a prototype assured compliance system called Safeguard. Safeguard is an independent avionics system for small to midsize UAS designed to be a very simple system that can constrain a drone's operation. Safeguard is designed to monitor and enforce conformance to a set of rules defined prior to flight, including geo-limits (e.g., geospatial stay-out or stay-in regions) and altitude constraints. Sections III and IV elaborate the rationale and suggested criteria for an assured compliance system. The Safeguard system and its current status are described in Section V. Then Section VI describes steps taken to produce an assured geofence.

III. The Case for Assured Geofencing

Safety and certification concerns about drones often focus on two hazards: failure to avoid other aircraft and loss of command and control capability (or link). Both hazards can have catastrophic consequences, but they are not the only hazards affecting UAS. The fly away hazard, wherein a drone flies into no-fly zones (prohibited areas or beyond its authorized range), also has the potential to cause catastrophic harm to people, other aircraft, and property. Operation in no-fly zones can result from intentional or uninformed actions on the part of the pilot or from loss of system capability (e.g., loss of position data, autopilot failure, or loss of command and control capability). Incidents where drones are encountered in no-fly zones are steadily increasing, despite regulatory restrictions and numerous campaigns to educate drone pilots [18]. For example, the U. S. Forest Service reported 15 drone intrusions complicating aerial firefighting efforts in multiple states in the 2014-2015 wildfire seasons [19]. More recently, drones have struck manned aircraft in airspace that was not authorized for drone operation [20-22].

The principle assertion for the concept of assured geofencing is as follows: if a drone's operation can be kept within areas that minimize safety risk to other aircraft, people and property, then some airworthiness requirements or

operational limitations may be simplified or relaxed [11]. Functionality to maintain operation of a drone within authorized airspace is key, and its implementation must be assured. That is, integrity of the data used for navigation and the reliability of compliance monitoring automation must be assured. The core functionality includes estimating the position of the drone, tracking its proximity to virtual perimeters established for all geo-limits, and signaling when action must be taken to ensure that the drone does not breach a perimeter. The concept is not new, and a number of drones already come equipped with various forms of this capability [23-25]. The reliability of existing geofencing systems, however, is unknown.

Geofences used in commercial UAS today are typically implemented via software, in conjunction with the autopilot, sharing the same sensors, computer processor, and operating system. Most commercial autopilots are not built in compliance with aviation safety standards. Those that are, command a much higher price. Some of the most popular autopilots are open source, raising concerns about software assurance. In addition, the lack of independence can lead to single points of failure: if the autopilot system fails, the geofence fails. Further, most geofencing functions rely on global navigation satellite systems (GNSS) for positioning. Space-based radio frequency systems share many common failure modes due to issues such as multipath, signal attenuation, and shadowing, creating another single point of failure. The use of GNSS alone cannot guarantee accurate, reliable geo-referenced positioning essential for maintaining geo-containment.

The known challenges of reliable geofencing beg the question of what makes a good geofence, one that could be assured to aviation safety standards while meeting industry requirements for size, weight, power and cost.

IV. Criteria for Assuring Geofencing Functions

Two recent research studies investigated airworthiness requirements of low-risk drone operations [26]. The studies focused attention on drones in the middle of the multidimensional spectrum of drone types; that is, drones with attributes and capabilities exceeding the criteria in Part 107, but without the design or operational capabilities to comply with the airworthiness standards for commercially-operated manned aircraft. The studies examined the degree to which existing airworthiness requirements might be suitable for drones, and suggested requirements for systems and equipment novel to drones. Geofences were among the novel systems addressed in the studies.

The studies examined airworthiness requirements for drones used in scenarios with defined geo-limits: (1) in agricultural applications, where authorized operation is limited to uninhabited fields, and (2) in rural cargo delivery, where authorized operation is limited to uninhabited corridors. Both studies identified the need for functionality to assure the drone stays within authorized geo-limits, and assumed that functionality could be realized through on-board automation. Failure of geofencing functionality was considered catastrophic since it could result in a drone entering an area or behaving in a manner that is hazardous to other aircraft or persons on the ground. The August 2, 2010 incident where an MQ-8 Fire Scout became unresponsive to commands during testing and entered protected airspace around Washington, D.C [27] is a well-known example.

While there is no explicit requirement in the FARs that states that an aircraft should only operate where authorized, Part 91 includes operational requirements stating where aircraft should not operate (e.g., in restricted areas per Part §91.133) [28]. That is, pilots of manned aircraft are responsible for avoiding operation in restricted or other hazardous areas. For drones, requirements are needed for that same functionality when implemented in an avionics system. As part of the research studies, an initial set of criteria were suggested for developing a system to detect and avoid transgression of specified geo-limits. A research prototype was created and tested and, as a result, the criteria has been updated, as follows:

1. Geo-limit data integrity – means to check the validity and timeliness of the geo-limit data should be provided (e.g., validity of the data sources and suitability of data for use by the detection algorithms).
2. Position data availability and accuracy – sufficient data should be available to estimate the drone's position at all times. The accuracy should be sufficient to ensure that the drone will not breach the geo-limits.
3. Situational awareness – awareness of the drone's position relative to the geo-limits should be maintained.
4. Detection – the means of detecting boundary violations should be able to monitor all defined boundaries and recognize impending violations in sufficient time for action to avoid breaching the geo-limits.
5. Pilot alerting – quick acting means should be provided to alert the pilot in command (or auto-pilot), if action is required. When human pilots are in command, timing thresholds for alerts should consider the time needed to transmit and process data, for annunciation, and for human response.
6. Avoidance – means of avoiding breach of any geo-limits should be sufficient to ensure the drone remains within the established geo-limits at all times. Latency and availability of any command and control datalink should be considered, if pilot action is required.

7. Collateral damage – events wherein release of high energy parts outside the geo-limits may constitute a hazard should be considered in detecting impending violations.
8. Interference – performance should not be degraded by any form of interference including, but not limited to electromagnetic interference from systems internal or external to the drone.
9. Dependencies – dependencies on external infrastructure, such as the Global Positioning System (GPS) or systems internal to the drone (e.g., autopilot, power, and datalinks) should be considered in evaluation of reliability.

The list is intended as a starting point for determining performance requirements for automated reliable systems. Safeguard was developed with these considerations/criteria in mind [29]. The next section describes the current Safeguard system prototype and results of initial flight testing to help validate requirements for such compliance-assurance systems.

V. Prototype System

Safeguard is designed to monitor and enforce conformance to a set of rules defined prior to flight (e.g., geo-limits and altitude constraints) for small to midsize drones. Safeguard is an independent system (i.e., not embedded with the autopilot or any other system) designed to be easily ported to virtually any drone: home-built, hobby, or commercial. Figure 1 shows the current version of the system installed atop a multi-rotor vehicle.

Fig. 1 Current Safeguard Prototype.

The Safeguard architecture is shown in Figure 2. As shown in the figure, Safeguard receives information on geo-limits and other performance parameters prior to flight through the base station. Position, navigation, and timing (PNT) data is provided by a high-grade GPS/INS/altimeter system contained within Safeguard and independent of the drone's PNT system. An alternate PNT system may be part of Safeguard depending on the integrity or assurance level requirement for the mission. Position and proximity of the drone to the conformance criteria (e.g. geo-limits) is monitored continuously throughout the flight by boundary violation prediction and detection algorithms. These algorithms produce warning signals and flight termination signals that can be fed to down-stream systems to initiate actions, as defined by the operator. Down-stream systems may be, for example, the remote pilot, the auto-pilot, or power termination relays.

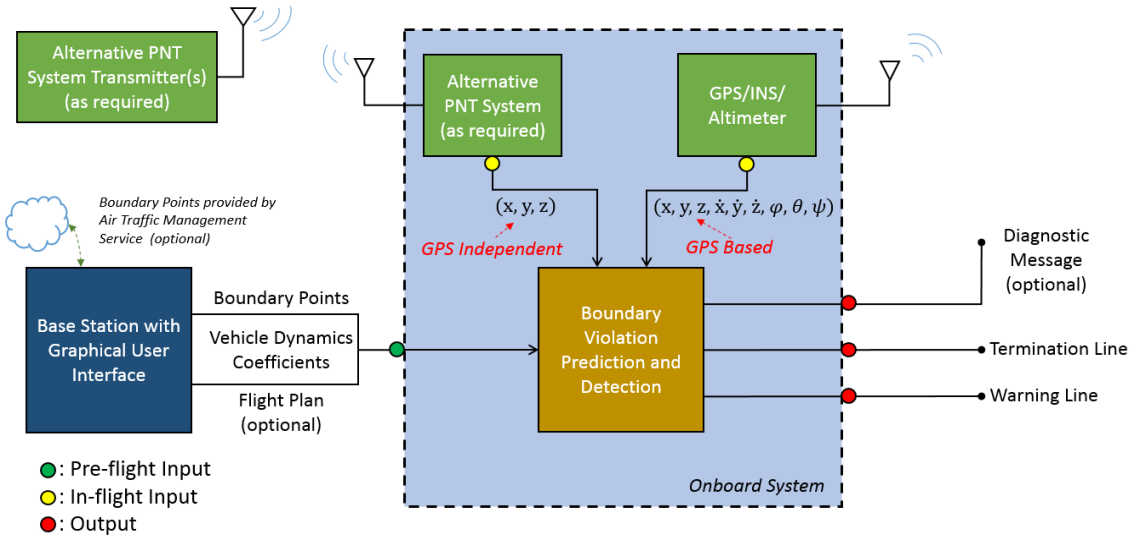

Fig. 2 System Architecture [29].

The core functionality for boundary violation prediction and detection is determining the position of the drone relative to the geo-limits defined for the operation. Geo-limits may be defined as stay-in regions (authorized or safe areas of operation) or stay-out regions (restricted or hazardous areas of operation). Only one stay-in region is allowed per operation, but there may be many stay-out regions within the stay-in region. All geo-limits must be defined in terms of polygons with a floor and ceiling altitude. Polygons may be any shape or size, as long as the polygon is closed (i.e., the first and last vertices are coincident and edges do not cross). Points that define geo-limits are captured and loaded prior to flight through the base station. Those points may be defined by the user or provided by a service such as UTM, in accordance with industry standards for similar data (see Section VI-A).

Safeguard establishes three boundaries for each geo-limited region: a hard boundary, a terminate boundary, and a warning boundary, shown in Figure 3 for simple square areas. The hard boundary (in red) represents a defined geo-limit that should never be breached.

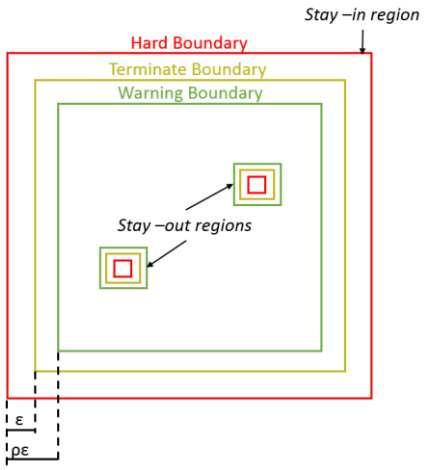

Fig. 3 Boundaries for Geospatial Constraint Conformance [29].

The terminate and warning boundaries (in yellow and green respectively) indicate buffers based on the drone's state and dynamics characteristics. The warning boundary defines the points when a signal will be issued to the remote pilot, autopilot, or other control system indicating that the drone is approaching a terminate boundary. The warning signal allows the drone to attempt a contingency maneuver to avoid violating a geo-limit. The sizes of the buffers vary dynamically depending on the maximum distance (ε) that the drone could travel if power to the motors is cut off. The warning boundaries change as a function of ε multiplied by a tunable scale factor ρ ($\rho>1$). The scaling factor provides

operators flexibility with respect to desired proximity to geo-limits. Establishing terminate boundaries at a distance ε from hard boundaries ensures that flight termination will prevent violations of geo-limits. Further specifications of the derived boundaries and how they are dynamically defined can be found in [29].

Once initialized, Safeguard continuously monitors the drone's position to predict breaches of the defined boundaries. In flight, Safeguard can communicate system status information to the pilot, communicate warning signals to the control authority to allow contingency maneuvers, and can issue flight termination signals. When the termination signal is used, various flight termination policies and mechanisms can be considered and these will depend on the drone type, operation/mission, risk tolerance, and operational environment. For example, flight termination for a multi-rotor may simply involve cutting power to the motors, if crashing is acceptable in the operational environment (e.g., flight over an uninhabited field). Termination policies and mechanisms may be more complex when operating over or near populated areas and beyond visual line-of-sight (e.g. ceasing propulsion and deploying a parachute). In any case, termination policies must comply with regulatory requirements for the operation.

VI. Safety Assurance

Developing an assured system capable of meeting aviation grade safety standards was a research goal from the outset. Meeting this goal has entailed strategies to overcome the technical challenges described in Section III, plus actions to provide confidence and evidence that Safeguard could eventually meet conventional aviation safety requirements. One action central to establishing assured functionality was to develop Safeguard's software in compliance with NASA's software safety standards for Class B software (i.e. mission critical, non-human space flight) [30]. Compliance with the Class B standards required detailed documentation of the system and software requirements, risk assessment, and extensive verification and quality assurance activities similar, but not identical, to those required in DO-178C [6]. While taking such a step is not typical when developing a research prototype, going through a safety standard compliance process was useful to identify and minimize defects and provide a sound step towards claims regarding assurance.

The following subsections describe additional steps taken to address safety assurance, including data integrity, formal verification, and flight testing.

A. Data Integrity

The criteria listed in Section III includes two important considerations for data: integrity of the stored data used to define the conformance criteria (e.g. geo-limits), and sufficiency of the in-flight data used to estimate the position of the drone during flight. Assuring that both types of data are valid, timely, and secure is essential to support high assurance claims for the system as a whole.

1. Geo-limit Data Considerations

Because of the functional importance of valid and correct geo-limits, all such data for Safeguard are captured, processed, and transferred in accordance with appropriate Data Processing Assurance Levels (DPALs) for similar types of data used on manned commercial transport aircraft (e.g., navigation and airport mapping databases) [31]. The polygons that represent the hard boundaries for no-fly zones represent the most complex and vulnerable data to errors. Fortunately, there are several established industry standards for assuring the content and quality of these types of data. These standards were established for commercial transport aircraft that utilize similar geospatial data for navigation and situation awareness systems, where the likelihood of corrupt, erroneous, or misleading data must be very low. The procedures for assuring integrity of the Safeguard databases leverage guidance from six such standards [31-36].

Standards for processing data that are to be used onboard aircraft are defined in [31]. Any data to be acquired, processed, and loaded onto an aircraft system should comply with this standard, as well as guidance provided in [33]. The primary purpose of these standards is to assure that (a) the data provided meets all of the requirements for its intended use, and (b) the data has not been altered or corrupted since origination. Seven quality characteristics are established in [31] wherein evidence must be provided to support the claims of the designer with respect to meeting the system's data quality requirements. These are:

1. Accuracy – The degree of conformance between the estimated or measured value and its true value
2. Resolution – The number of units or digits to which a measured or calculated value is expressed and used
3. Assurance Level – The degree of confidence that a data ***element*** is not corrupted while stored or in transmission
4. Traceability – The degree that a system or a data product can provide a record of the changes made to that product and thereby enable an audit trail to be followed from the end-user to the data originator

5. Timeliness – The degree of confidence that the data is applicable to the period of its intended use
6. Completeness – The degree of confidence that all of the data needed to support the intended use is provided
7. Format – The structure of data elements, records and files arranged to meet standards, specifications or data quality requirements

For Safeguard, the requirements for six of these are given in [32-34] and are assumed to be sufficient for most missions. Characteristic #3 is referred to as the Data Processing Assurance Level (DPAL) and, per the standard, may be one of three levels (1, 2, or 3); with "1" being the highest degree of confidence. Typically, the DPAL will correspond to the Design Assurance Level (DAL) associated with the software that uses the database [6]. For example, a DPAL of "1" corresponds to a DAL of "A" and "B" (that is, software whose anomalous behavior could contribute to a catastrophic or hazardous failure condition).

The DPAL requirement for pre-loaded data in Safeguard will likely vary across missions and operating environments based on the level of risk deemed acceptable with respect to violating prescribed constraints (e.g., hard boundaries). For research and development purposes, the most stringent is assumed to be required (DPAL 1). The method to achieve DPAL 1 will depend on whether the data originates locally via a process managed and performed by the operator, or the data is provided as a service from an authorized source. Details on both of these methods will be published separately.

2. Positioning Data Considerations

For the Safeguard system to reliably and independently perform its function, accurate and timely position estimates are critical. Positioning data for manned aircraft are available from a myriad of independent systems such as high integrity GNSS receivers, very high frequency omnidirectional range (VOR) stations, distance measuring equipment (DME), tactical air navigation (TACAN) stations, and high quality inertial navigation systems (INSs). Redundancy is typically employed to mitigate potential failures and ensure continuous operation. Unfortunately, low altitude operations, common for many drone applications, can cause ground-based radio systems to be unobservable due to line of sight issues. Moreover, most drones employ sensors selected for low weight and cost, which typically results in a lower performing positioning system consisting of a GNSS receiver and inertial measurement unit (IMU). Because of the (low-cost, low-weight, small-size) drivers, techniques to mitigate known GNSS vulnerabilities are not implemented (e.g. signal attenuation, jamming, and multipath). As a result, the available sensors cannot be relied upon for safety critical applications, even if methods such as those in [37] are employed.

For Safeguard to effectively and independently monitor proximity to geo-limits, the performance of its positioning system must be better than the performance of positioning system embedded within the drone. With this in mind the current prototype includes a high-grade high-performance GPS/IMU/altimeter system. In addition, for environments where degraded GPS performance is expected (e.g. urban canyons), Safeguard has been designed to include a secondary independent positioning system so that multiple sources of positioning data can be used to mitigate accuracy and availability problems that can occur with GPS. For current testing and proof-of-concept, a Locata® [38] local positioning system was chosen as the alternative PNT system. Locata® uses a network of small, ground-based transmitters to provide radio-positioning signals, independent of GPS. Other alternate PNT systems will be evaluated in the future.

It should also be noted that for cases where the COTS drone does have a high-grade high-performance GNSS/IMU sensor, Safeguard functionality may be embedded within the COTS system to provide some degree of assured functionality with respect to geo-limit compliance monitoring. However, the loss of independence will lead to some loss of assurance/integrity.

B. Algorithmic Design and Formal Verification

It is common for geofencing systems to model safe and unsafe areas using polygons. Typically, these are two dimensional polygons on the surface of the earth, which also include a minimum and a maximum altitude. The software components that compute whether a given position is inside or outside of these regions often involve mathematics that is complicated enough to raise concerns about its correctness. For these reasons, a formal approach to algorithm design is desired, whereby the following three properties hold

1. It should be possible to trace software requirements to formal specifications and formally proved correctness properties.
2. It should be possible to easily relate these specifications to their implementations in code.
3. The behavior of the code should correspond to the formal specifications, even in the presence of floating point errors.

Completely ensuring each of these properties is impossible, but the approach used to formally verify the core logic in Safeguard shows that it is possible to develop geofencing functions where each of these properties is formally addressed.

Using the prototype system design process to demonstrate the method, it is possible to trace geofence requirements for the software in the Safeguard system to formal specifications and formally proved correctness properties. Indeed, the main functions in Safeguard are formally specified, and many of their core sub-functions have been formally proven to be mathematically correct when implemented with infinite precision real numbers (not floating-point numbers). For instance, Safeguard uses ray casting as one method for containment determination, and one sub-function of this algorithm checks whether a ray, which is cast from a given point, crosses a given polygon edge. This function, along with many others, has been formally proved correct in an interactive theorem prover [39]. It should be noted here, however, that a complete formal proof of a ray casting method is quite difficult and most likely corresponds to the nontrivial Jordan Curve Theorem, although a formal proof of this theorem has been completed before.

In the Safeguard system, it is possible to directly relate formally specified algorithms to their implementations in code through visual inspection. Most of the functions in Safeguard have identically named implementations in both the specification language and in code. This is intentional and has the purpose of ensuring that a one-to-one correspondence exists between the implementation and its functional requirement.

Finally, the Safeguard software has been checked by a large suite of test cases, consisting of 500 polygons and 200 test-points per polygon. This test suite helps verify that the implementations in code of the Safeguard containment functions agree with their formal specifications. This is accomplished through model animation, whereby the formal specifications are evaluated using semantic attachments for non-computable functions, e.g., square root. Since the code level implementations of algorithms use floating-point numbers and their specifications use infinite precision numbers, exact agreement is impossible to ensure. However, numerous properties are checked for each test case. For instance, for any input position on which the specification and code disagree on whether it is contained in a given region, it is checked that the point lies very close to one of the polygon edges. This therefore checks not whether the floating-point arithmetic correctly implements infinite precision arithmetic (which it does not), but rather it checks whether any floating-point errors do not cause problems that practically negate benefits of the formally proved correctness properties of the specifications.

As Safeguard illustrates, it is indeed possible to improve the assurance that geofencing algorithms correctly implement system requirements. While a complete formal assurance of this statement is elusive, formal specifications, formal proofs, and model animations allow one to trace behavior from requirements through executable code.

C. Hardware Considerations and System Flight Testing

The flight test plan for Safeguard includes five primary objectives: (1) demonstrate correct performance in nominal conditions across different vehicle types, including small rotary-wing and fixed-wing UAS; (2) demonstrate correct functionality during periods of degraded GPS performance, including loss of GPS; (3) evaluate alternate termination strategies; (4) demonstrate integration with the UTM system and services; and (5) demonstrate correct functionality when operating beyond visual line-of-sight. All of these tests are over and above what was done to verify the software in accordance with the Class B certification process.

Flight tests to achieve the five objectives have been and are being conducted using Safeguard on a number of UAS platforms. The current Safeguard prototype uses the Class B-compliant software and COTS hardware components. The unit has been installed and flown on numerous multi-rotor platforms. For this testing, the warning line was connected to the vehicle's autopilot, while the terminate line was either not connected, or connected to either a system that forces the vehicle to land or a mechanism which simply disconnects power to the UAS' motors. While an action such as the discontinuation of propulsion is intended as a termination action for some Safeguard applications, many of the flight tests were conducted with the termination line triggering an action to land to prevent perpetual damage to test vehicles. As discussed previously, based on the vehicle's mission environment and risk tolerance, various termination policies and mechanisms can be employed and are being considered for future tests. Efforts to reduce the overall size, weight, and power (SWAP) of the unit and enhance ruggedness are also underway.

Recent tests have focused on flight test objectives 2, 4 and 5, using an octo-copter platform flown on the campus of NASA Langley Research Center. Test flights were designed to verify intended functionality in a variety of missions and operational scenarios, as well as differing flight plans and altitudes. The number, complexity and types of boundaries for no-fly zones were also varied to cover a wide range across flight tests. To force Safeguard into scenarios with degraded GPS, flight operations were conducted in close proximity to buildings and other large structures to simulate urban environments. Additionally, many flight tests were conducted with connectivity to UTM services. The

results of flight tests in two of the operational scenarios are described in detail in [40]. A more comprehensive set of results will be published in the future.

VII. Closing Remarks

No matter the size of the drone or its mission, its behavior depends inescapably on the integrity of the data it uses and the reliability of the avionics systems that process that data to affect the operation of the aircraft. This gives rise to a predicament in that many components for commercial drones are based on COTS technology that is relatively inexpensive, which helps foster market growth. However, COTS components often lack sufficient assurance of their safety and reliability, which restricts expansion of regulations that allows market growth. Requiring compliance of all drone systems with conventional aviation safety standards would be untenable from a cost perspective. One possible approach to support expanded operations in the short term is to strategically use aviation-grade components (i.e., components meeting approved aviation standards) to provide an assured bound on the behavior of systems that lack safety assurance evidence. If the size and complexity of the safety-critical systems can be minimized, the certification costs may not be cost-prohibitive. The Safeguard system is a specimen system that limits the airspace in which a drone can operate, using a very simple design. A number of steps including data integrity checks, formal verification, and standards compliance have been taken to support safety assurance of the system. Research is on-going to validate the system and cost assertions for certification.

VIII. References

[1] Hughes, David, "Information Technology: This Changes Everything…" *Aviation Week & Space Technology*, December 21/28, 1998.

[2] United States Government, (undated), Title 14 Code of Federal Regulations, Part 107, Small Unmanned Aircraft Systems, [Online], Available: http://www.ecfr.gov/cgi-bin/textidx?SID= ae4f72f9345bad959a0d89dc3084918f&mc=true&node=pt14.2.107&rgn=div5

[3] European Aviation Safety Agency, 'Prototype' Commission Regulation of Unmanned Aircraft Operations, 22 August 2016, [Online], Available: https://www.easa.europa.eu/system/files/dfu/UAS%20Prototype%20Regulation%20final.pdf

[4] Society of Automotive Engineers (SAE), "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment", SAE ARP 4761, December 1996.

[5] Society of Automotive Engineers (SAE), "*Guidelines for Development of Civil Aircraft and Systems",* SAE ARP 4754A, *December 2010.*

[6] RTCA Special Committee 205, "Software Considerations in Airborne Systems and Equipment Certification," RTCA Document DO-178C, December 2011.

[7] RTCA Special Committee 180, "Design Assurance Guidance for Airborne Electronic Hardware," RTCA Document DO-254, April 2000.

[8] RTCA Task Force 4, "Executive Summary of the Final Report of the RTCA Task Force 4 Certification", February 26, 1999.

[9] Gilligan, Margaret, "Statement of Margaret Gilligan, Associate Administrator for Aviation Safety, Federal Aviation Administration, Before the Committee on Transportation and Infrastructure, Subcommittee on Aviation, On Building a 21st Century Infrastructure for America", February 15, 2017.

[10] Hayhurst, Kelly J., Maddalon, Jeffrey M., Morris, A. Terry, Neogi, Natasha A., and Verstynen, Harry A., "A Review of Current and Prospective Factors for Classification of Civil Unmanned Aircraft Systems," NASA/TM-2014-218511, August 2014.

[11] Hayhurst, K., Maddalon, J., Neogi, N., Verstynen, H., "A Case Study for Assured Containment," International Conference on Unmanned Aircraft Systems", Denver, CO, June 2015.

[12] Sha, Lui, Goodenough, John B., and Pollack, Bill, "Simplex Architecture: Meeting the Challenges of Using COTS in High-Reliability Systems," Crosstalk, pp. 7-10, 1998.

[13] Kopardekar, P., "Unmanned Aerial System (UAS) Traffic Management (UTM): Enabling Low Altitude Airspace and UAS Operations," NASA Ames Technical Memorandum, 2014.

[14] ASTM WK53403, "New Practice for Methods to Safely Bound Flight Behaviors of UAS Containing Adaptive Algorithms, 2017.

[15] JAR-DEL-WG6-D.04, "JARUS guidelines on Specific Operations Risk Assessment (SORA)", 2017.

[16] European Aviation Safety Agency, "Introduction of a regulatory framework for the operation of drones", September 2016.

[17] European Aviation Safety Agency, "Study and Recommendations regarding Unmanned Aircraft System Geo-Limitations", Notice of Proposed Amendment 2017-05 (A), May 2017.

[18] Federal Aviation Administration, UAS Sightings Report, August 9, 2017, [Online], Available: https://www.faa.gov/uas/resources/uas_sightings_report/

[19] Carpenter, Kaari, "Unauthorized Drones Near a Wildfire can Cost and Kill", U.S. Forest Service Blog, August 10, 2016, [Online], Available: https://www.fs.fed.us/blogs/unauthorized-drones-near-wildfire-can-cost-and-kill

[20] UAS Vision, “Drone Hits Boeing 737 Approaching Buenos Aires Airport”, posted November 14, 2017, [Online], Available: https://www.uasvision.com/2017/11/14/drone-hits-boeing-737-aproaching-buenos-aires-airport/
[21] CBS News, “A first in Canada: Drone collides with passenger airplace above Quebec City airport”, posted October 15, 2017, [Online], Available: http://www.cbc.ca/news/canada/montreal/garneau-airport-drone-quebec-1.4355792
[22] Furfaro, Danielle, Celona, Larry, and Musumeci, Natalie, “Civilian drone crashes into Army Helicopter, New York Post, September 22, 2017, [Online], Available: https://nypost.com/2017/09/22/army-helicopter-hit-by-drone/
[23] Stevens, Mia N., Rastgoftar, Hossein, and Atkins, Ella, “Specification and evaluation of geofence boundary violation detection algorithms”, 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, June 2017.
[24] Pratyusha, P.L. and Naidu, V.P.S., “Geo-Fencing for Unmanned Aerial Vehicle,” International Journal of Computer Applications (0975-8887), 2013.
[25] Ardupilot, “Simple geofence,” [Online] http://copter.ardupilot.com/ wiki/ac2_simple_geofence/.
[26] Hayhurst, K. J., Maddalon, J. M., Neogi, N. A., and Verstynen, H. A., “Design Requirements for Unmanned Rotorcraft Used in Low-Risk Concepts of Operation,” NASA/TM– 2016-219345, November 2016.
[27] Bumiller, Elisabeth, “Navy Drone Violated Washington Airspace”, The New York Times. August 25, 2010, [Online], Available: http://www.nytimes.com/2010/08/26/us/26drone.html
[28] United States Government, (undated), Title 14 Code of Federal Regulations, Part 91, General Operating and Flight Rules, [Online], Available: https://www.ecfr.gov/cgi-bin/text-idx?SID=0169cfade3dbac1dfa9f2504a7acd6af&mc=true&node=pt14.2.91&rgn=div5div5
[29] Dill, E., Young, S., and Hayhurst, K., “Safeguard: An Assured Safety Net Technology for UAS”, IEEE/AIAA Digital Avionics Systems Conference (DASC), Sacramento, CA, September 2016.
[30] National Aeronautics and Space Administration, “Software Assurance Standard,” NASA Technical Standard NASA-STD-8739.8, NASA, 2004.
[31] RTCA Special Committee 217, “Standards for Processing Aeronautical Data,” RTCA Document DO-200B, June 2015.
[32] RTCA Special Committee 217, "User Requirements for Aerodrome Mapping Information," RTCA Document DO-272D, November 2015.
[33] RTCA Special Committee 217, “User Requirements for Terrain and Obstacle Data," RTCA Document DO-276C, November 2015.
[34] RTCA Special Committee 217, “Interchange Standards for Terrain, Obstacle and Aerodrome Mapping Data," RTCA Document DO-291C, November 2015.
[35] Federal Aviation Administration, “Acceptance of Aeronautical Data Processes and Associated Databases,” Advisory Circular AC-20-1538, April 2016.
[36] RTCA Special Committee 217, “Standards for Aeronautical Information,” RTCA Document DO-201A, April 2000.
[37] Farrell, J., “GNSS Aided Navigation and Tracking,” American Literary Press, 2007.
[38] Rizzos, C., “Locata: A Positioning System for Indoor and Outdoor Applications Where GNSS does not Work,” Proceedings of the 18th Association of Public Authority Surveyors Conference, 2013.
[39] Munoz, Cesar, and Narkawicz, Anthony, “Formalization of Bernstein Polynomials and Applications to Global Optimization”, Journal of Automated Reasoning, Volume 51, Issue 2, pp. 151-196, August 2013.
[40] Young, Steven D, Dill, Evan T., Hayhurst, Kelly J., and Gilabert, Russell V., Safeguard, Progress and Test Results for a Reliable Independent On-board Safety Net for UAS, IEEE/AIAA Digital Avionics Systems Conference (DASC), St. Petersburg, FL, September 2017.