SAFEGUARD

Progress and Test Results for A Reliable Independent On-board Safety Net for UAS

Steven D. Young, Evan T. Dill, Kelly J. Hayhurst, and Russell V. Gilabert
Safety-Critical Avionics Systems Branch
NASA Langley Research Center
Hampton, VA, 23681

***Abstract*—As demands increase to use unmanned aircraft systems (UAS) for a broad spectrum of commercial applications, regulatory authorities are examining how to safely integrate them without compromising safety or disrupting traditional airspace operations. For small UAS, several operational rules have been established; e.g., do not operate beyond visual line-of-sight, do not fly within five miles of a commercial airport, do not fly above 400 ft above ground level. Enforcing these rules is challenging for UAS, as evidenced by the number of incident reports received by the Federal Aviation Administration (FAA). This paper reviews the development of an onboard system – Safeguard – designed to monitor and enforce conformance to a set of operational rules defined prior to flight (e.g., geospatial stay-out or stay-in regions, speed limits, and altitude constraints). Unlike typical geofencing or geo-limitation functions, Safeguard operates independently of the off-the-shelf UAS autopilot and is designed in a way that can be realized by a small set of verifiable functions to simplify compliance with existing standards for safety-critical systems (e.g. for spacecraft and manned commercial transportation aircraft systems). A framework is described that decouples the system from any other devices on the UAS as well as introduces complementary positioning source(s) for applications that require integrity and availability beyond what can be provided by the Global Positioning System (GPS).**

This paper summarizes the progress and test results for Safeguard research and development since presentation of the design concept at the 35th DASC (2016). Significant accomplishments include completion of software verification and validation in accordance with NASA standards for spacecraft systems (to Class B), development of improved hardware prototypes, development of a simulation platform that allows for hardware-in-the-loop testing and fast-time Monte Carlo evaluations, and flight testing on multiple air vehicles. Integration testing with NASA's UAS Traffic Management (UTM) service-oriented architecture was also demonstrated.

Keywords—geofencing; Unmanned Aircraft System; formal methods; UAS Traffic Management (UTM)

I. INTRODUCTION

UAS have diverse designs and performance capabilities, with corresponding diversity in safety risk. Some differences in UAS are obvious, such as the size, weight, or configuration of the aircraft. Other differences are more subtle, such as the frangibility of the structures and materials or the provenance of the components. While these vehicle differences affect safety risk, the operational context—especially the environment in which a UAS flies—also affects safety risk for UAS, more so than for conventional commercial aircraft. UAS have the potential for causing catastrophic harm to people, other aircraft, and property when they operate in prohibited areas or beyond their authorized range. 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). According to the FAA, reports of inappropriate drone operations have increased dramatically over the past two years, with the FAA receiving over 100 such reports each month [1].

To mitigate these risks, the FAA provides information on restrictions and requirements in effect at specific locations and has implemented Part 107 of the Federal Aviation Regulations for small UAS operations. Most of the requirements in Part 107 specify operational limitations in lieu of levying vehicle-specific requirements; for example, constraining operations to visual line-of sight, daylight-only, 400 ft altitude above ground level, and maximum groundspeed of 87 knots [2]. In conventional aircraft, the onboard pilot is primarily responsible for conforming to operational limitations. Ensuring conformance to operational limitations for UAS presents some challenges due to dependence on UAS operators who may lack sufficient aviation safety knowledge and experience and/or on automation that does not have the same safety provenance of conventional aviation automation. This has given rise to a need for novel systems and equipment to assure safe UAS operations. Geofencing systems are one example.

In this paper, geofencing refers to constraining UAS flight to an airspace defined by geographical and altitude limits (providing a stay-in function) and/or preventing a UAS from entering prohibited airspace (providing a stay-out function). Geofencing is not a novel concept for UAS [3][4]. Geofencing has been recognized as fundamental to mitigating safety risk by the European Aviation Safety Agency (EASA) in their proposed rulemaking for UAS [5], [6], by the American Society for Testing and Materials (ASTM) F38 working group [7] and in plans for UAS traffic management (UTM) [8]. While home-built UAS and those used by hobbyists may not have a geofencing system, a number of commercial UAS do come with some degree of built-in geofencing capability [9], [10].

Sightings of UAS operating in prohibited areas speaks to the need for more reliable geofencing, and there are known technical factors that affect the reliability of embedded geofencing systems. For example, geofences used in commercial UAS are typically embedded with the autopilot, using the same computer processor and operating system. For autopilots that lack the provenance of conventional avionics systems (e.g., open source autopilots), establishing confidence in the reliability of the geofence can be challenging [11], [12]. 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.

To rely on geofencing to prevent unintended flight beyond safe areas, geofencing systems should meet some level of assurance for reliability and dependability. Likewise, automation to ensure conformance with other operational limitations, such as speed limits and altitude constraints, should be reliable. This paper identifies some preliminary considerations for developing a reliable onboard system for geofencing and describes the development and flight testing of a prototype system intended to meet high design assurance levels. This work is expected to extend to enhanced functionality for ensuring conformance to a broader set of operational limitations.

II. Deriving Considerations for Geofencing Functions

Reliable geofencing functionality depends critically on having correct and timely information on all geospatial boundaries and limitations relevant to a UA's operation [6]. The data aspects are recognized as an important role for UTM systems [8]. Reliable geofencing functionality also depends on having an avionics system that can act dependably on that data to ensure conformance with applicable geo-limitations. In this section, we focus on the latter.

A recent study of airworthiness requirements for UAS identified failure to stay within authorized geo-limitations as a potentially catastrophic hazard [13]. To mitigate the hazard, an independent, onboard geofencing system was proposed. Design and performance criteria were suggested for developing the system as a starting point for establishing risk-based requirements for geofencing systems. The criteria for geofencing as shown here could be easily extended for other conformance criteria:

C.1 *Data integrity*. A means to check the validity, timeliness, and security of the geo-limitation data should be provided (e.g., validity of the data sources and suitability of data for the detection algorithms).

C.2 *Position data availability and accuracy*. Sufficient data should be available to estimate the UAS position at all times. The accuracy should be sufficient to ensure that the UAS will not breach the geo-limits.

C.3 *Situational awareness*. Awareness of the UAS position relative to the geo-limits should be maintained.

C.4 *Detection*. The means of detecting boundary violations should monitor all defined boundaries and recognize impending violations in sufficient time for action to avoid breaching the geo-limits.

C.5 *Pilot alerting*. Quick acting means should be provided to alert the pilot in command if pilot action is required. Timing thresholds for alerts should consider the time needed to transmit and process data, for annunciation, and for human response.

C.6 *Avoidance*. The means of avoiding breach of any geo-limits should ensure the UAS 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.

C.7 *Collateral damage*. Events wherein release of high energy parts (e.g., from midsize rotorcraft) outside the geo-limits may constitute a hazard should be considered in detecting impending violations.

C.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 UAS.

C.9 *Dependencies*. Dependencies on external infrastructure such as GNSS or systems internal to the UAS (e.g., autopilot, power, and datalinks) should be considered in evaluation of reliability and security.

A prototype geofencing system, called Safeguard, was developed with these considerations in mind. The following sections describe the system and how it relates to the design criteria, considerations for assuring performance, and recent flight test results.

III. A Prototype Geofencing System

Safeguard is an independent geofencing system to support UTM for small to midsize UAS [14]. Safeguard is a simplex architecture [11], [15]: a very simple system that constrains the geospatial behavior of a more complex system—the autopilot—that is more difficult and costly to certify. The core functionality is based on determining whether the position of the UAS is inside or outside a set of polygons that represent the geo-limitations for the operation. Geo-limitations may be defined as stay-in regions (authorized or safe areas) or stay-out regions (restricted or hazardous areas). As per C.1 and C.2, the geo-limitation data and position data for the UAS are critical for an effective geofencing function.

In Safeguard, three boundaries are established for each stay-in and stay-out region: a hard boundary (the geo-limit), a terminate boundary, and a warning boundary. These are illustrated in Figure 1 for simple square regions. These boundaries help provide situational awareness with respect to the proximity of the UAS to geo-limits (per C.3). The hard boundary (in red in Figure 1) is a user-defined polygon representing a safe area that should never be breached. Points that define hard boundaries are loaded prior to flight. Polygons may be any shape or size, as long as the polygon is closed (i.e.,

the start-point and end-point are coincident and no boundaries intersect). There may be only one stay-in region per operation, but many stay-out regions.

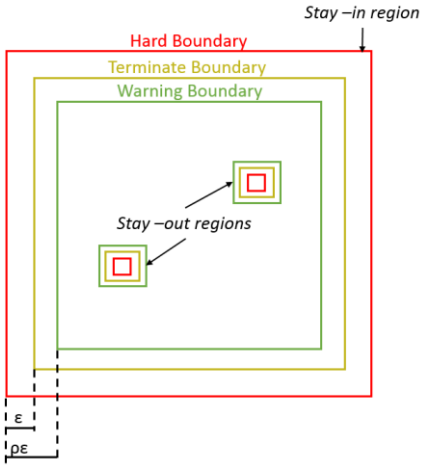

Figure 1. Boundaries for Geospatial Constraint Conformance.

The terminate and warning boundaries (in yellow and green respectively) indicate buffers based on the vehicle's state and its' aerodynamic characteristics. The warning boundary defines the points when a notice will be issued to the remote pilot, autopilot, or other systems indicating that the vehicle is approaching a terminate boundary. The warning allows the UAS to attempt a contingency maneuver to avoid flight termination. The size of the buffers depend on the maximum distance (ε) that the UAS could travel if flight termination were initiated. The warning boundaries dynamically change as a function of ε multiplied by a tunable scale factor ρ (ρ>1). The scaling factor provides operators flexibility with respect to desired proximity to terminate boundaries, including consideration of collateral damage from release of high energy parts (per C.7 Collateral Damage).

If the UAS crosses the terminate boundary, loss of control or unrecoverable fly away is assumed and flight termination can be initiated to avoid breaching the hard boundary. Establishing terminate boundaries at a distance ε from hard boundaries ensures that flight termination (e.g. by cutting power) will prevent violations of geo-limits (per C.6).

The system architecture for Safeguard is shown in Figure 2. Prior to flight, stay-in and stay-out boundary coordinates, vehicle dynamics parameters, and an optional flight plan are loaded. In the current prototype, this data is loaded by the operator, but may be automated through a service provider or UTM system in the future.

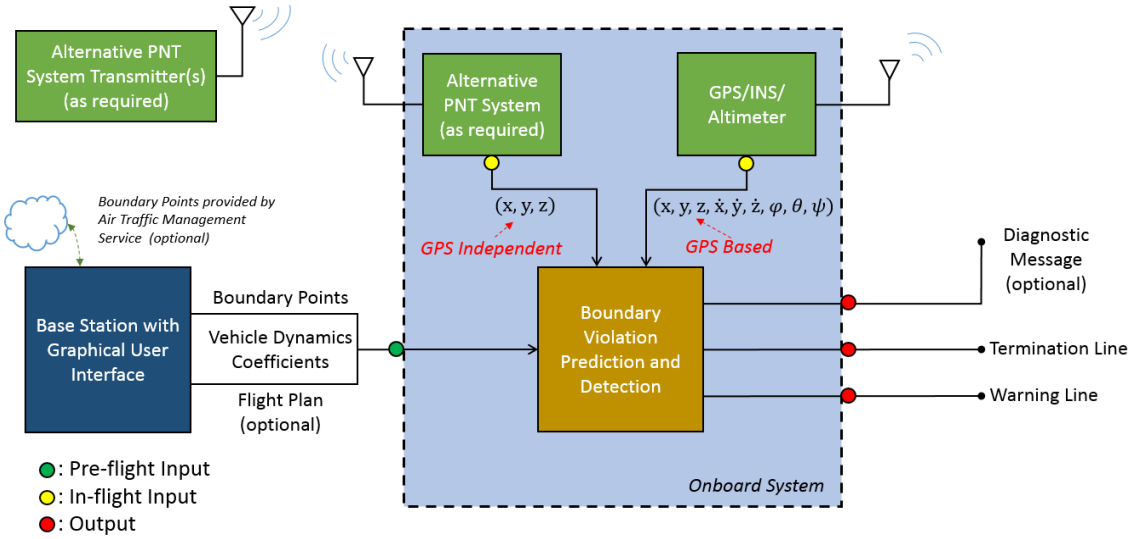

Figure 2. System Architecture.

Because of the functional importance of valid and correct geo-referenced hard boundary points, all such data are captured, processed, and transferred in accordance with appropriate Data Processing Assurance Levels (DPAL) [17] for similar types of data used on commercial aircraft (e.g., navigation data). This provides a check on data integrity per C.1.

Once initialized, Safeguard continuously monitors the UAS position to predict breaches of the defined boundaries (per C.2). Multiple sources of positioning data are used to mitigate accuracy and availability problems that can occur with GPS. In addition, two different methods are used to detect impending boundary violations per C.4: ray casting and winding numbers. With each update, the warning and terminate lines are set to either "compliant" or "violation". These lines usually are connected to two separate elements of

the UAS. The warning line is typically connected to a system that has control authority of the UAS, such as the autopilot or a contingency management function such as ICAROUS [16], to allow that system to initiate a contingency maneuver to avoid flight termination. Because off-the-shelf systems may fail, the terminate line is connected to a separate high integrity termination mechanism. Appropriate termination mechanisms will depend on specified limitations within the operational area (stay-in region), but may include options such as triggering parachute deployment or cutting power to the motors.

IV. Assurance Approach

Developing a highly reliable geofencing system capable of meeting aviation grade safety standards was a goal from the outset. Meeting such a goal entailed strategies to overcome the technical challenges described in Section I, plus assurance measures to provide confidence and evidence that Safeguard could meet design and performance requirements. Having an independent system coupled with a simple architecture that could facilitate compliance (and ultimately certification) with aviation safety standards was considered crucial to making a reliable system at a price point suitable for the current UAS market.

After validating initial research prototypes of Safeguard as described in Dill et. al. [14], the Safeguard system was taken to the next level of technical readiness by developing the Safeguard software in compliance with NASA's safety standards for Class B software (for mission critical, non-human space flight) [18]. Compliance with the Class B standards required detailed documentation of the system and software requirements, risk assessment, and extensive verification and quality assurance activities. According to [19], "software assurance assists in risk mitigation by minimizing defects and preventing problems and, through its activities, enables improvement of future products and services." This level of assurance may not be necessary for all UAS missions, but may be necessary when operational risk assessment determines that staying within specified geo-limitations is safety-critical. In those cases, Safeguard can provide a safety net for conventional geofencing systems.

The following subsections provide additional detail relevant to assurance for the three main elements of the Safeguard system: (1) the positioning system, (2) the boundary database, and (3) the boundary monitoring and violation detection software.

A. Positioning System Performance

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 GNSS, 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 can cause ground-based radio systems to be unobservable due to line of sight issues. Moreover, most UAS employ sensors selected for low weight and cost, which typically results in a lower performing positioning system consisting of a GNSS receiver and a lower-grade inertial measurement unit (IMU). Because of GNSS vulnerabilities such as signal attenuation, jamming, and multipath, the available sensors cannot be relied upon for safety critical applications, even if methods such as those shown in [20] are employed. For Safeguard to be an effective monitor, the performance of its positioning system must be better than the performance of the positioning system embedded within the UAS. As such, Safeguard has been designed to operate with a secondary independent positioning system. For current testing and proof-of-concept, a Locata® [21] local positioning system was chosen as the alternative PNT system (APNT). Locata® uses a network of small, ground-based transmitters to provide radio-positioning signals, independent of GPS. Other APNT systems will be evaluated in the future.

B. Boundary Database Integrity

The second essential input to the detection algorithm is the set of geospatial and other constraints that are specified pre-flight and loaded onto the Safeguard unit. Of these, the most complex and vulnerable to errors are the polygons that represent the hard boundaries (i.e., no-fly zones). 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 probability of failure must be very low. The procedures for assuring integrity of the Safeguard databases leverage guidance from six such standards [17][22][23][24][25][26].

Standards for processing data that are to be used onboard aircraft are defined in [17]. Any data to be acquired, processed, and loaded onto an aircraft system should comply with this standard, as well as guidance provided in [25]. The primary intents are to assure that (a) the data provided meets all of the requirements for its intended use, and (b) data has not been altered or corrupted since origination. Seven quality characteristics are established in [17] 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 [17][22][23][24] 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 [27]. 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).

As with positioning system performance, it is expected that the DPAL requirement for pre-loaded data in Safeguard will 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, we assume the most stringent will 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 a certified source. Details on both of these methods will be published separately.

C. Verification of Boundary Detection Software

Assuring the accuracy and behavior of the algorithms for detecting proximity to and violation of geo-limitations is central to a reliable system. To support the creation of highly-assured algorithms, the boundary detection and violation prediction functions in Safeguard were developed and verified using formal methods [28]. In particular, the Prototype Verification System (PVS) was used to specify and verify the algorithms. Using PVS enabled the identification of corner cases, such as problematic geometries, increasing the robustness of the algorithms. Additional verification activities included extensive testing both in simulation and flight.

As it is not practical to terminate a multitude of vehicles, a UAS simulation capability developed at NASA Langley is being leveraged. This simulation enables testing and refinement of the software functionality in an accelerated and benign fashion. The simulator allows for accelerated Monte Carlo evaluations across a span of input uncertainties and other failure modes without having to physically crash any UAS. Monte Carlo evaluations have been conducted for each formally verified function.

V. Flight Testing and Results

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.

Flight tests have been and are being conducted using Safeguard on a number of UAS platforms. The current Safeguard prototype, shown in Figure 3, uses the Class B software and commercial-off-the-shelf (COTS) hardware. This unit has been installed and flown on numerous multi-rotor platforms. For this integration, the warning line was connected to the vehicle's autopilot, while the terminate line was 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. Based on the vehicle's mission environment and risk tolerance, various other termination mechanisms can be employed and are being considered for future tests. Efforts to reduce the overall size of the unit and enhance ruggedness are also underway.

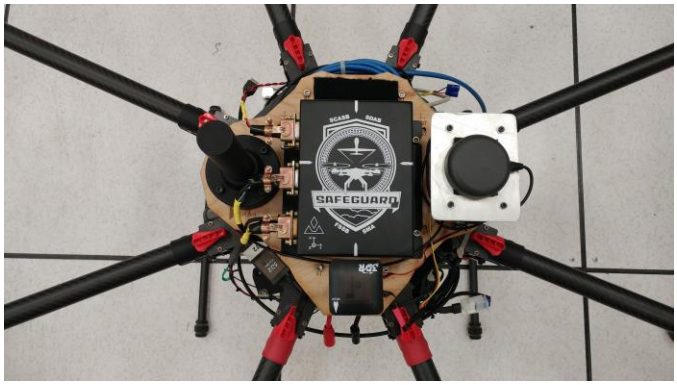

Figure 3. Current Safeguard Prototype

Recent tests have focused on flight test objectives 2, 4 and 5, using an octocopter platform flown on the campus of NASA LaRC. 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, many of the flight operations were conducted in close proximity to buildings to simulate urban environments. Additionally, all flight tests were conducted with connectivity to UTM services. The results of flight tests in two of the operational scenarios are given in the remainder of this section. A more comprehensive set of results will be published in the future.

The first example operational scenario was designed to capture output signals from Safeguard while encountering the boundaries of a simple rectangular stay-in region. Figure 4 shows this region covering part of the NASA LaRC campus. For this test Safeguard was not integrated with any other onboard system. Thus, no mitigating actions were taken to prevent excursions beyond the boundary. Various flight plans

were chosen such that the UAS crossed over the same boundary numerous times. This forced Safeguard's internal state to vary and trigger all of its possible output modes. Example results of this type of test are shown in Figure 5 where the flight data is colored to indicate Safeguard's internal state and associated output signal: "safe" (green), "warning" (yellow), "terminate' (red), and "violation" (black). Through analysis of data collected over a series of these tests, Safeguard was found to have functioned as designed with zero false alarms or missed detections. Furthermore, Figure 5 illustrates instances of the dynamically changing size of the warning and terminate buffers. Per the system design, warnings are triggered at different distances from the hard boundary due to variations in the platforms velocity when approaching the boundary.

Figure 4. Stay-in Region for Flight Tests above Langley Boulevard at NASA LaRC (Operational Scenario 1).

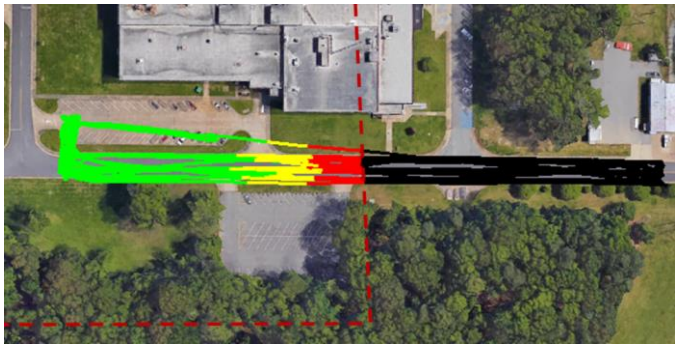
Figure 5. Flight Results from Operational Scenario 1.

A second example operational scenario, depicted in Figure 6, evaluated performance using a single rectangular stay-in region along with multiple complex stay-out zones. The stay-in region was established to constrain flight to within close proximity of the intended operation above Dryden Street at NASA LaRC. The various stay-out regions were established to prevent the vehicle from flying into (or above) buildings or above intersections on the road. To prevent excursions beyond these operational boundaries, Safeguard's warning line was setup to trigger a "bounce" maneuver (i.e. cease forward momentum, then move away from the encountered boundary), and the terminate output was configured to initiate an auto-land. To visualize how the warning and terminate boundaries dynamically change during flight, Figure 7 shows a graphical comparison of each boundary location during two different instants of a flight at dissimilar speeds (left: 2 m/s, right: 6 m/s) and at an altitude of 30 m above ground level (AGL).

Figure 6. Stay-in and Stay-out Regions for Flight Tests above Dryden Street at NASA LaRC (Operational Scenario 2).

Figure 7. Warning and Terminate Buffers Dynamically Changing During Operational Scenario 2 at 2 m/s (left) and 6 m/s (right).

Within the previously described environment, two different test situations were conducted. Both of these attempted to accomplish the same autonomously-flown flight plan at an altitude of 30 m AGL and a velocity of 8 m/s. This flight plan began at the south-western end of Dryden Street and traversed above the road in a north-eastern direction for approximately 200 meters. This flight plan was purposefully designed to encounter the stay-out zone located at the intersection of Langley Boulevard and Dryden Street (Figure 7).

For the first situation, all systems were left in a functional state and the results shown in Figure 8 were produced. As the UAS began to approach the stay-out zone above the intersection, Safeguard issued a warning signal and the multi-rotor "bounced" away from the no-fly zone. Then, the aircraft re-attempted to complete its flight plan and was "bounced" off of the no-fly zone two additional times at slower velocities due to the loss of momentum after the initial encounter with the warning boundary. It can be seen that the vehicle's initial encounter with the warning boundary occurred at a further distance from the stay-out zone than subsequent encounters. This is due to the vehicle's reduced velocity after each bounce and re-try. Safeguard functionality during this test performed as expected, warning in sufficient time to allow the bounce maneuver to be executed and the vehicle to re-assess whether to continue the flight plan.

Figure 8. Flight Results from Operational Scenario 2 with Warning Triggering "Bounce" Maneuvers.

The second situation was designed to emulate a fly-away scenario as would occur with an autopilot failure. This allows for testing Safeguard's flight termination signal. Example results of this test are depicted in Figure 9 where the autopilot ignores input commands, including Safeguard's warning signal, and continues to travel toward the no-fly zone. Once the terminate boundary was reached, Safeguard triggered a vehicle termination thereby preventing a boundary violation in the presence of an autopilot failure.

Figure 9. Flight Results from Operational Scenario 2 with Simulated Autopilot Failure and Terminate Signal Triggering Land Maneuver.

VI. Future Work

Work remains to be done to develop Safeguard to a level of technical readiness for certification and commercial use. While much attention has been paid to the software and software assurance, little effort has been devoted to hardware ruggedization. Additional work is also needed to reduce size, weight, power and cost (SWAP-C) for some vehicles. Based on an analysis of the computing and sensor requirements for Safeguard, it is believed that a ruggedized version could be produced that is approximately 1"x2"x3", with a weight of 8oz and a nominal power consumption of 300mA. Steps are underway to achieve this via partnering discussions with manufacturers of similar devices, as well as using NASA in-house expertise.

Additional work is also needed to refine the estimation of the minimum safe separation distance (MSSD) from hard boundaries to ensure that flight termination prevents any violation of no-fly zones. To improve the MSSD estimation, work is underway to better characterize vehicle dynamics. We believe that loading a small set of vehicle dynamics parameters may be sufficient to characterize each unique vehicle, rather than loading a high-fidelity aerodynamic model. While the current set of vehicle dynamics have been shown to adequately describe multiple UAS, additional testing is needed for a broader range of aircraft. Data captured during flight termination events may also help refine the MSSD estimation.

Environmental factors, especially wind, can affect the MSSD estimate. One approach being explored is over-bounding wind speeds and directions based on forecasts prior to flight. Flights may also be restricted from takeoff if winds exceed a certain threshold.

Conducting test flights on different UAS types in various environmental conditions (e.g., in wind conditions and in rugged terrain with GPS dropouts) will be necessary to meet flight test objective 1. These tests will also be used to confirm design considerations C.8 (interference) and C.9 (dependencies on other systems). Flight test data is being used to verify that the system works as designed, and to validate data uncertainty models used in Monte Carlo simulations. Flight test data along with data gathered during simulations and NASA standards compliance evaluations are intended also to help develop a safety case for this type of technology.

VII. Closing Remarks

Despite federal regulations on UAS operations and campaigns to inform and educate UAS pilots, incidents of inappropriate UAS operation continue. These incidents give rise to a need for solutions to assure safe operations for small to midsize UAS. Assured geofencing systems that enforce conformance to geo-limitations or other operational limitations have the potential to help advance integration of UAS safely in the near term. Having an aviation-grade geofence can potentially reduce regulatory requirements on some UAS operations and enable operations in areas that are currently prohibited. The Safeguard system is one such development that could help facilitate the expansion of UAS operations by reliably keeping drones out of no-fly zones and other potentially hazardous environments.

References

[1] Federal Aviation Administration, UAS Sightings Report, [Online] https://www.faa.gov/uas/resources/uas_sightings_report/

[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] Atkins, E., "Autonomy as an enabler of economically-viable, beyond-line-of-sight, low-altitude UAS application with acceptable risk," AUVSI Unmanned Systems, 2014.

[4] Pratyusha, P.L. and Naidu, V.P.S., "Geo-Fencing for Unmanned Aerial Vehicle," International Journal of Computer Applications (0975-8887), 2013.

[5] European Aviation Safety Agency, 'Prototype' Commission Regulation on Unmanned Aircraft Operations, 22 August 2016.

[6] European Aviation Safety Agency, "Study and Recommendations regarding Unmanned Aircraft Systems Geo-Limitations", EASA/NAA Task Force Report, 02 September 2016.

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

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

[9] DJI, "DJI Introduces New Geofencing System for its Drones," press release, 18 November 2015, [Online] https://www.dji.com/newsroom/news/dji-fly-safe-system

[10] Ardupilot, "Simple Geofence", product documentation, [Online] http://copter.ardupilot.com/ wiki/ac2_simple_geofence/

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

[12] Stevens, Mia, and Atkins, Ella, "Multi-Mode Guidance for an Independent Multicopter Geofencing System," 16th AIAA Aviation Technology, Integration, and Operations Conference, Washington, DC, June 2016.

[13] 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.

[14] Dill, E., Young, S., and Hayhurst, K., “Safeguard: An Assured Safety Net Technology for UAS”, IEEE/AIAA Digital Avionics Systems Conference (DASC), September 2016.

[15] 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.

[16] Consiglio, M., Munoz, C., “ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems”, IEEE Digital Avionics Systems Conference, 2016.

[17] RTCA Special Committee 217, “Standards for Processing Aeronautical Data,” RTCA DO-200B, June 2015.

[18] National Aeronautics and Space Administration, “Software Assurance Standard,” NASA Technical Standard NASA-STD-8739.8, NASA, 2004

[19] European Aviation Space Agency, “Introduction of a regulatory framework for the operation of drones,” Advance Notice of Proposed Amendment, 2015-10, July 31, 2015.

[20] Farrell, J., “GNSS Aided Navigation and Tracking,” American Literary Press, 2007.

[21] 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.

[22] RTCA Special Committee 217, "User Requirements for Aerodrome Mapping Information," RTCA Document DO-272D, RTCA, November 2015.

[23] RTCA Special Committee 217, “User Requirements for Terrain and Obstacle Data," RTCA Document DO-276C, RTCA, November 2015.

[24] RTCA Special Committee 217, “Interchange Standards for Terrain, Obstacle and Aerodrome Mapping Data," RTCA Document DO-291C, RTCA, November 2015.

[25] Advisory Circular, “Acceptance of Aeronautical Data Processes and Associated Databases,” AC-20-1538, April 2016.

[26] “Standards for Aeronautical Information,” RTCA Document DO-201A, RTCA, April 2000.

[27] “Software Considerations in Airborne Systems and Equipment Certification,” RTCA Document DO-178C, RTCA, December 2011.

[28] Monin, J., “Understanding Formal Methods,” Springer-Verlag, London, 2003.