# Establishing the Assurance Efficacy of Automated Risk Mitigation Strategies

Natasha Neogi, Steven Young and Evan Dill

*NASA Langley Research Center, Hampton, VA, 23666, USA*

**Abstract**

Verification and validation of increasingly autonomous aviation systems is a major challenge. Traditional techniques for the assurance of high-confidence, safety-critical systems are not equipped to handle the complexity, uncertainty, and lack of predictability inherent in non-deterministic systems. Techniques such as run time monitoring, formal methods, and testing and simulation have been applied to some effect, but it is difficult to properly assess the success of such measures. The authors propose the concept of *Assurance Efficacy* to address this gap. Assurance Efficacy is seen as a parameter, criteria, or perspective by which to evaluate, identify and explore safety risk mitigation strategies and operational assurance architectures. Validation of the utility of this concept through flight testing is a first step in determining its potential role in assessing the overall safety of complex, increasingly autonomous systems that cannot be fully assured in the design phase.

## I. Introduction

Emerging aviation markets such as uncrewed aerial systems (UAS) cargo delivery, urban air mobility (UAM), and commercial space launch systems are catalyzing a transformation of the national airspace system. This transformation is partially powered by two developing technologies: aircraft electrification and autonomy. In 2021, Morgan Stanley predicted that the global market for UAM, which is likely to employ electric vertical takeoff and landing (eVTOL) vehicles with increasingly autonomous systems, will be over $1 trillion [1]. By comparison, the global airline aviation market was valued at $818 billion in 2019 (pre-Covid-19 levels) [2]. However, this figure was revised downwards to $1 trillion in an update to their report in 2021. The analysts at Morgan Stanley "made significant reductions to our adoption curve in the early and medium-term years to allow for a greater 'margin of safety' given our understanding of the certification landscape in the U.S. and Europe" [3]. This draws specific attention to the challenge of the certification and assurance of these systems.

Verification, validation (V&V) and certification remain key challenges to the fielding and usage of increasingly autonomous (IA) systems in safety-critical contexts. Multiple barriers are associated with these challenges [4]. These include, but are not limited to, the following: (1) Most traditional V&V approaches and methods are insufficient for assuring advanced IA systems (e.g., due to the complexity of the underlying software), (2) Existing certification criteria, processes, and approaches do not address some of the unique characteristics of advanced IA systems (e.g., the potential for non-deterministic behavior), (3) Existing safety standards and requirements, which are focused on assuring the safety of aircraft passengers and aircrew, must be extended to consider the safety of third parties who are not involved in the operation (e.g., pedestrians on the ground and others). Additionally, the global civil aviation industry estimates that 27,000 new pilots will be needed by the end of 2021, and up to 264,000 new pilots over the coming decade [24]. This may be a precursor to a shift in the locus of control for safety-critical decision making from the pilot to autonomous/automated functions.

There are several promising paths forward to enabling the assurance of increasingly autonomous systems: (1) run time assurance and run time verification (RTA/RTV) techniques; (2) formal methods and analysis; and (3) advanced simulation, testing, and evaluation, among other methods. Of increasing interest to the aviation community is the deployment of RTA/RTV frameworks during flight in order to monitor systems that cannot be adequately assured during design via traditional means. The goal of these frameworks is to reduce risks during operations that cannot easily be addressed using conventional design techniques. The RTA approach does become part of the design, but its effect on safety assurance does not manifest until operations. In other words, it allows risks related to hazards previously addressed at design time (or unknown at design time) to be alleviated using operational means and methods.

More formally, the utility of RTA/RTV frameworks is predicated on: (1) the ability to clearly specify the (safety) properties of interest to be monitored, (2) the ability to implement minimally invasive monitors independent of the system under observation, (3) the ability to demonstrate that the specified safety properties are correctly monitored, (4) the ability to assure both the monitors and the monitoring architecture via conventional means to the required design assurance level (DAL) to achieve the desired safety properties, and (5) the assurance level of the monitor(s) must be higher than the assurance level of the system being monitored. Additionally, for runtime monitoring to be truly effective, there must be a mitigation action that can be executed once the monitor observes that a safety property is violated (or predicted to be violated), and the monitor and mitigation action *must be assured to the desired design assurance level of the function being monitored*. The combination of the monitor and mitigation action *must guarantee* that the system is always brought back into a safe state and/or kept from transgressing into an unsafe state.

This paper focuses on evaluating the ability (or difficulty) to assure the performance of an operational mitigation intended to alleviate the risk posed by hazards that may traditionally have been tackled using conventional design methods. A key factor when deciding between these options is the cost, time, and uncertainty associated with the required V&V activities. The authors suggest this factor can be addressed using the concept of ***Assurance Efficacy (AE)***. The concept can be applied to any operational mitigation approach; however, in this paper, the authors focus on automated approaches. Section II performs a brief survey of the current state-of-the art in UAS certification as well as runtime assurance approaches. Section III defines and further explains the AE concept. Section IV discusses an architecture that is being used to evaluate the AE of several mitigation strategies drawn from a real-world system that NASA is implementing for flight test. Section V generalizes the notion of using AE to make design decisions that will establish safety properties for a system. Highlights will include the qualitative approach to AE developed through the analysis of an example architecture along with some of the challenges and pitfalls associated with developing a quantitative approach. Section VI will then summarize the conclusions of this research as well as avenues for future work.

## II. Background and Definitions

### A. Background

#### *1. UAS Integration into the NAS*

In order to gain access to the National Airspace System (NAS), a vehicle must possess an airworthiness certificate as well as have operational approval. An airworthiness certificate is traditionally achieved via means of a type certificate for the vehicle along with being in a safe operating condition. Operational approval for most general aviation operations occurs via 14 CFR Part 91 [16] which occurs along with the FAA's Air Traffic Organization authorizing access to the NAS.

14 CFR Part 107 [18] authorizes the class of small UAS (weighing under 55 lbs.) to access the NAS so long as operations are performed during daylight hours within visual line of sight of the pilot in command, outside of proximity to people or infrastructure not party to the operation, and in Class G airspace. In general, for UAS over 55 lbs, access to the NAS is granted on a case-by-case basis, either through waivers to 14 CFR Part 107 or 14 CFR Part 135 [17]. However, the "Operation of Unmanned Aircraft Systems Over People" final rule allows for the routine operation of small UAS over people and routine operation of small UAS at night under certain circumstances. This rule eliminates the need for these specific operations to receive individual Part 107 waivers from the FAA. The rule was published in the Federal Register on January 15, 2021, and the effective date of implementation is April 21, 2021 [19].

However, for vehicles which do not fall under the rubric of Part 107 or the FAA "Operation of Unmanned Aircraft Systems Over People" final rule, a form of certification will likely be required to access the NAS. Even the final rule sets out four classes of vehicles and requires increasing degrees of assurance rigor to be applied to each class, with class IV of the rule invoking the standard certification process. Of note, there are several certification challenges for UAS (small or otherwise) that may only be addressed at run time (i.e., during flight). These include unanticipated vehicle interactions, unanticipated external interactions, mission management decisions with flight-critical consequences, untested system modes, and autonomous decision-making and control [5]. Additionally, there are several initial efforts underway in the FAA's Integration Pilot Program (IPP) regarding UAS cargo-carrying operations that have received an air operator's certificate under Part 135 and have had airworthiness requirements waived for the duration of these initial operations.  For instance, FAA issued the first Part 135 air carrier certificate for single pilot operations to Wing Aviation, LLC for UAS operations in April 2019, and later issued Wing a Standard Part 135 air carrier certificate to operate a UAS in October 2019 [28].

### *2.   Increasingly Autonomous and Highly Autonomous Systems (IA/HA)*

This sub-section summarizes some of what has been discussed and written on the subject of autonomy in recent years – particularly with regards to AAM concepts such as UAM. Within these, there is an implied push to IA/HA and a significant change to the roles and responsibilities of humans (e.g., operators and fleet managers vs. pilots and controllers). However, the community has recognized that safety assurance will be the enabler or the constraint to such operational concepts. Some in the community assume that traditional methods will suffice; others realize that these methods would delay implementation and make many types of operations cost-prohibitive. While there is a great deal of existing knowledge, expertise, and standards associated with automation (e.g., auto-land, auto-pilot, and auto-braking systems), there remains a leap when considering automation that crosses the blurred line into IA/HA. In this paper, the authors consider this line to be defined by the inclusion of safety-critical decision-making functions that can impact the overall mission through changing the vehicle state or flight plans/maneuvering without human intervention (and oversight in some cases).

### *3.   Runtime Assurance and Verification (RTA/RTV)*

Runtime assurance refers to the process of assuring a system's behavior and/or performance properties during the operational phase of the system lifecycle, as opposed to during its design phase [6]. Most runtime assurance techniques involve: (1) a system under observation, which cannot be assured during design to the required DAL; (2) a monitor, which expresses the required property that is to be guaranteed over the system's operation and is assured to a higher assurance level than the system being observed; (3) an assured mitigation function which either steers the system state into the safe set or recovers it into that set; and (4) the runtime assurance architecture, which provides the decision logic (or functionality) that switches the system to the assured system when/if necessary [8].  A common version of this architecture is referred to as a Simplex architecture [7] and is now generally housed under the broader term of Runtime Assurance (RTA). RTA has a long history in computer science and computer engineering applications; however, they are only recently being formalized for consideration in regulatory airworthiness approval processes [9].

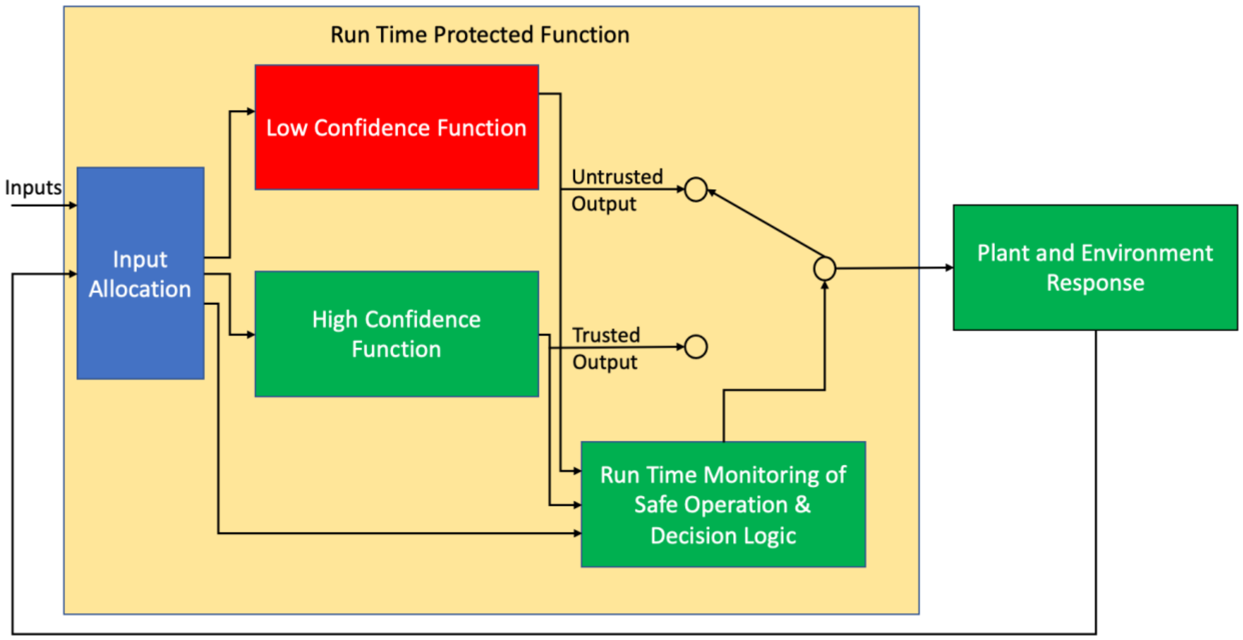
Run Time Protected Function
Low Confidence Function
Inputs
Input Allocation
Untrusted Output
High Confidence Function
Trusted Output
Plant and Environment Response
Run Time Monitoring of Safe Operation & Decision Logic

**Figure 1:  Run Time Architecture (Simplex)**

A closely related term and concept is runtime verification (RV). RV (a.k.a., RTV) is an area of formal software verification methods that studies the dynamic analysis of software execution traces against formal specifications of software.  Runtime monitors detect and respond to property violations at runtime and have the potential to enable the safe operation of safety-critical software-intensive systems that are too complex to formally verify or fully test [11].

High-assurance RTV provides a guaranteed level of safety when the software functions under observation cannot be verified by conventional means. The runtime verification monitor takes the logical specification $\phi$ and the execution trace $\tau$ of the state information for the system under observation and evaluates whether $\tau$ satisfies $\phi$. If $\phi$ is not satisfied, then the monitor produces an alert. However, using only RTV techniques does not address the challenge of how to design an assured mitigation function that will return the system to a state where it satisfies $\phi$ [12].

The authors seek to develop and advance the concept of AE such that it may be useful to designers when making decisions regarding the above gaps and challenges. For example, assuming that the RTV property $\phi$ to monitor has been specified and implemented correctly, the effect of the mitigation function on the execution trace can be evaluated with respect to its efficiency in returning the system to the desired set of states. This includes the task of smoothing the transition between the control outputs of the untrusted function and the (trusted) mitigation function such that a hazardous state is not engendered by this switch. Less formally, if the hazardous state(s) being monitored are known, the risk associated with those hazardous states and the risk reduction achieved by executing a given mitigation action can potentially be evaluated.

Furthermore, the AE concept can provide a means of evaluating the degree of difficulty of employing various mitigation strategies (and their attendant logical architectures) in terms of buying down residual risk in an operational context. This includes assessing the degree of uncertainty in bounding unexpected behavior, as well as the effect on cost/benefit and time to implement versus using traditional design-time methods. This is a novel approach as run time assurance is not usually considered for certification credit in traditional design-time certification practices. Similarly, operational assurance is usually implemented through operational procedures, processes, and restrictions as well as safety management systems along with the training of personnel. The use of the AE concept marries the notion of taking design time requirements and monitoring them in runtime, and then assessing how well the mitigation actions taken reduce the risk associated with the violation of the desired properties (and their hazards).

## III. Assurance Efficacy (AE)

To describe the AE concept, several safety-oriented terms must first be introduced. For the purposes of this work, we use the definitions listed below, with some of these terms visually illustrated in Figure 2. Note that the definitions used here are taken from Ref. [13], and the authors acknowledge that there are multiple alternate definitions used by others, such as the FAA's definition of a hazard [14] as: "A present condition, event, object, or circumstance that could lead to or contribute to an unplanned or undesired event, such as an accident." The authors use the below definitions simply for the purpose of uniformity here and consider both definitions to be valid.

- Safety - the freedom from accidents or loss events (i.e., the absence of risk).
- Accident - an undesired and unplanned event that results in a loss (including loss of human life or injury, property damage, environmental pollution, etc.).
- Hazard - a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to an accident or loss event.
- Hazard Level - a quantification of hazard severity and hazard likelihood. A higher hazard level indicates the severity and/or likelihood is increased.
- Hazard Severity - the worst-case damage (or outcome) that could result from the hazard given the environment is in its most unfavorable state.
- Hazard Likelihood - the likelihood of a hazard occurring (during flight).
- Risk (a.k.a. Safety Risk) - a function of the hazard level combined with (1) the likelihood of the hazard leading to an accident and (2) hazard exposure time or duration. A higher risk indicates (1) and/or (2) are increased.

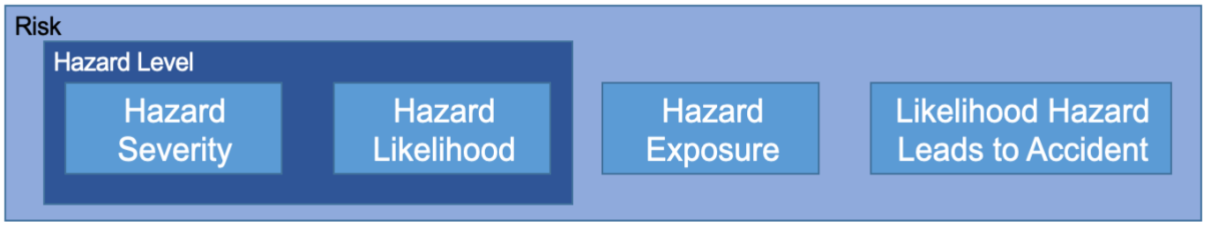

**Figure 2: Risk components [13]**

Traditionally, hazards are identified and assessed for their severity and likelihood early in the design process through a variety of means such as a functional hazard analysis [15]. Identified hazards are assessed and then either eliminated, mitigated, controlled, or accepted (typically based on a defined set of risk categories). These actions are performed in the design phase of the system. However, since not all hazards can be eliminated in the design phase, residual risks remain in the system as it transitions into the operational phase of the life cycle. Risks associated with hazards still present in the operational phase of the lifecycle are either mitigated, avoided, accepted, shared, or transferred. The concept of AE focuses on: (1) evaluating the residual risk associated with remaining hazards and (2) assessing the value, effectiveness, and cost/benefit of assurance strategies aimed at mitigating those risks during system operation. In general, AE seeks to help to decide how to move risks that are often accepted, shared, or transferred into the mitigated category (i.e., more of the total risk is mitigated).

For example, AE can be determined for a particular mitigation strategy executed in a given runtime assurance architecture with respect to a specific hazard (whose residual risk in the operational phase is non-zero). In this case, an AE measure can be defined as “the reduction in operational risk posed by the mitigated hazard as compared to the risk posed by the hazard in the unmitigated system.” Note that this definition of an AE measure is defined in the context of a specific mitigation strategy and operational assurance architecture. Depending on the detection strategy for the monitored property (e.g., conservative with respect to false alarms) and the decision logic (or function) used to switch to the mitigation strategy (e.g., sensitivity to uncertainty) a mitigation strategy can yield different assurance efficacies under different architectural paradigms and implementations.

Similarly, a mitigation strategy can act to mitigate multiple hazards. Thus, careful consideration must be taken before evaluating the AE of a mitigation strategy in terms of the ***overall or total risk***. Simply put, a mitigation may significantly reduce the risk of one hazard by a great deal, while reducing the risk posed by another hazard by a small amount. Similarly, another mitigation strategy may reduce the risk of both hazards by a moderate amount. Thus, the AE of the first mitigation is superior to the second mitigation with respect to the first hazard, but inferior with respect to the second. Decision logic (or functionality) in the architecture would then reflect that by selecting the first mitigation if the first hazard was detected. Similarly, the second mitigation would be selected in the case of the second hazard being present. However, if both hazards are present, it is not immediately clear which mitigation should be selected. If the value of total risk reduction with respect to both hazards is taken as a guide, erroneous safety conclusions can be made. For instance, consider that the second mitigation reduces the total system risk overall by a larger amount than the first mitigation in the presence of both hazards. It is natural to think that this would be the preferred strategy. However, consider that the first mitigation may be reducing the risk posed by the first hazard from catastrophic (e.g., fatalities) to major (e.g., no fatalities but a reduction in ability to perform safety roles). If the second hazard only posed a major risk to begin with (e.g., no fatalities but reduced margins), the reduction in risk achieved by the second mitigation with respect to that hazard may not be as much of a priority. Thus, careful consideration must be given when using the concept of AE for decision making outside of its clear definitions.

## IV. Case Study and Discussion

### A. Case Study Background

The simple example of an automated mitigation system for a small UAS performing a short haul operation (e.g., limited flight time) is considered. The set of automated (autonomous) mitigation actions include the following: (1) Return to base and land (i.e., return to launch (RTL)), (2) Land at nearest available prepared site, (3) Land at nearest unprepared site, (4) Flight Termination/Land Immediately, and (5) Hold in position. The hazards being mitigated include: (1) Loss of Containment in Approved Airspace (e.g., Flyaway), (2) Loss of Separation with People and Infrastructure on the Ground, (3) Loss of Control, and (4) Loss of Critical Systems (Communications, Navigation, Power etc.). The runtime assurance architecture created for testing is based on the system described in [20], with updates and additional details found in [25, 26]. This work also builds upon the research performed in [23].

### B. Architecture and Mitigation Strategies for Evaluating Assurance Efficacy

For testing, the functions shown in Figure 3 provide an implementation of an auto-mitigation capability that allows for assessing AE within the RTA/RTV framework. On-board functions are implemented as a flexible set of applications running on a Core Flight System (cFS)-based independent system (i.e., independent of the COTS autopilot) [20]. Applications utilize the cFS software bus (SB) to exchange messages with each other and with external systems or functions. A set of SB bridge applications (e.g., Safeguard Bridge and Mav Bridge) translate serial data streams from/to external systems [22]. This includes for example sensors and the COTS autopilot (A/P). A detailed description of each function shown in Figure 3 is provided in [26]. A high-level operational perspective is provided below.

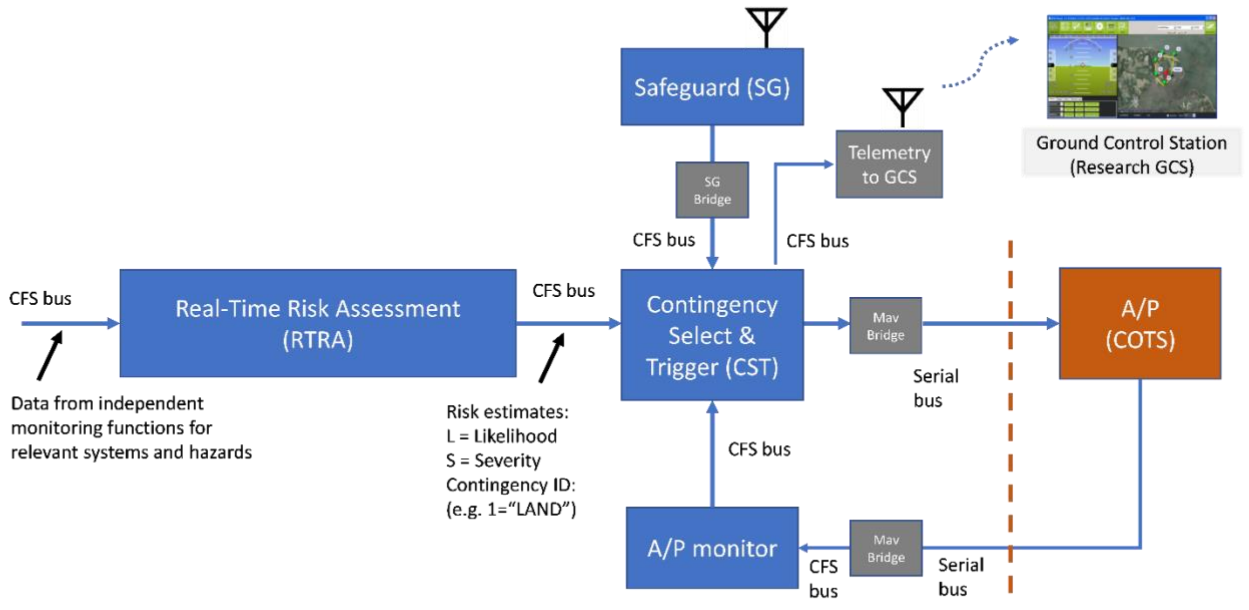

**Figure 3. Example onboard risk and contingency management automation in RTA/RTV framework [26].**

During flight, the Contingency Select and Trigger (CST) function is responsible for selecting from an available set of contingencies using a look-up table and inputs provided by the Real Time Risk Assessment function (RTRA) function, the Safeguard (SG) function, and the autopilot (A/P) monitor function. Selection is based on pre-defined logic that assures deterministic behavior within the operational limits of the COTS autopilot (shown on the right). Because the SG function has been developed and tested to the comparatively highest assurance level (NASA Class B safety-critical) [27], warnings from this unit are given priority by the logic. SG warns of impending violations of constraints (e.g., approaching a no-fly zone) and does so by producing two types of warnings: an early warning and a more urgent alert. Both occur before a violation occurs. As tested, early warnings are transmitted (downlinked) to the operator/pilot so that they may consider intervention (i.e., supervised mitigation). The CST function will request the A/P to switch to POS-HOLD mode until the operator/pilot decides how to proceed. Urgent alerts from SG require more immediate action. For this case, in addition to notifying the operator/pilot, CST will send a mode change command to the A/P to "LAND" immediately.

Additional CST logic is defined for situations where SG is not predicting a constraint violation (i.e., urgent alerts), but RTRA is estimating elevated risk (due to, for example, battery, engine, wind, or proximity to non-participants). Based on likelihood and severity estimates, as well as associated thresholds, RTRA may recommend A/P mode changes to mitigate the risk. Depending on the magnitude of the estimates and the type of hazard, various available mode change types may be requested (e.g., hover, land at current location, return to land at the launch location, and land at nearest available landing site). CST monitors the RTRA outputs and provides commands to the A/P accordingly.

Three precautions are taken to help assure deterministic and bounded behavior associated with the above logic. For both the constraint violations (i.e., warnings from SG) and elevated risk states (i.e., as estimated by RTRA), the A/P monitor data is checked to see if the desired CST command can be flown by the A/P given its current state. Also, a lookup table (used to determine the CST contingency actions) is stored pre-flight within CST and remains available in-flight to cross-check that RTRA-recommended contingencies match the expected mode changes/suggestions for the designed situations. Finally, as described previously, all functions to the left of the COTS A/P in Figure 3 are implemented independently and within the cFS framework [20].

The supervised mitigation strategy as described above also includes an aspect of the automated mitigation strategy in that the operator/pilot is informed of CST actions and may override the auto-mitigation if desired (as long as the telemetry and C2 links are functioning).

### C. Simulation and Flight Validation

Other than the COTS autopilot, the on-board software was developed to NASA Procedural Requirements (NPR) 7150.2C standards for Class B or C [27], with the Safeguard software also classified as safety-critical [21, 22]. As part of the standard, all on-board software related to controlling/commanding AP modes has undergone a series of bench tests prior to flight. As a result of the integration and the scenarios planned for flight testing, more than 90,000 possible software test cases were identified. Of these, 472 were identified as relevant failure cases. After removing duplicates, 92 unique cases remained. Each of these 92 cases were then tested using flight logs and a flight simulation tool. Results were checked against expected results. After fixing any exposed design errors, all cases were retested with results independently verified.

Flight testing of this and other developmental safety-related services, functions, and capabilities (SFC) began in the Fall of 2021 and are ongoing. A broad set of test scenarios have been defined covering four use-cases and a set of off-nominal situations. The CST software has undergone unit testing in order to scrutinize the function individually and independently for proper operation. The interactions with the SG Bridge, RTRA, APMon, and, indirectly, the autopilot were examined. Input conditions for the CST were generated via selection of a set of potential RTRA mitigation actions derived from the RTRA test case set. While the RTRA provides its own recommended mitigation action, the raw data provided to the CST (from RTRA) is synthesized and then used to come up with its own independent recommendation. This enables the CST to then compare both mitigations and determine if a common solution was reached. If both mitigations are in agreement (e.g., both RTRA and CST recommend the same mitigation command) then the CST publishes the commanded mitigation to the cFS once the AP monitor has been checked for the mitigation's feasibility. If the mitigation recommendations differ from one another, then the CST recommendation is published to the cFS. CST's interaction with the Safeguard bridge was also tested by simulating primary failures and warnings from seven main Safeguard warning categories. The CST mitigation actions were then compared against the Safeguard recommended mitigations. Additionally, the interaction between the CST and the AP monitor was exercised by having the CST receive a list of available modes from the AP monitor. The CST checks its mitigations against the list of available modes from the AP in order to determine feasibility of its mitigation before publishing it to the cFS. Preliminary results are provided in companion papers [25, 26], with a sample flight test scenario elaborated in full in [26]. A more comprehensive analysis of the final results will be published in the future via a journal paper or technical manuscript.

## V. Establishing Safety Using the Assurance Efficacy Concept

### A. Architectural Evaluation

While AE can be determined for a set of mitigation strategies given a runtime assurance architecture and a set of hazards, it is also possible to evaluate a set of architectures with respect to a prespecified set of mitigations and a given hazard. That is, for a given set of mitigations, it is theoretically possible to perform a design space exploration on the monitoring scheme and decision logic to identify the architecture that utilizes the mitigations to best effect. More concretely, given two architectures and a common set of mitigations designed to address a single hazard, it may be possible to determine which architecture is superior to the other ***with respect to AE***. Note that this is likely only one of the properties on which an architecture may be evaluated, and other properties such as ease of instrumentation or simplicity may be the dominating factor in architecture choice.

### B. Ordering in Assurance Efficacy

The AE of a set of mitigations can be ordered with respect to one another in the context of a given runtime assurance architecture and set of hazards. However, this ordering is a partial order as some mitigations may not be comparable to one another, namely if they do not mitigate a common hazard. Even if the mitigations are over a single hazard, it is possible that two mitigations reduce the risk by the same amount but may not do so ***in the same fashion***. That is, one mitigation may reduce the severity of the hazard, while the other may reduce the likelihood of that hazard. In this case, those mitigations are labelled ***incomparable*** with respect to one another, and the resulting order is a partial order. This is because there would be a great potential for drawing erroneous safety conclusions if the two mitigation strategies were labelled with the same AE.

### C. Qualitative versus Quantitative Approaches

Given the natural language definition of AE, there may be a tendency to try to quantify this parameter using numerical estimates of risk and risk reduction. Caution must be exercised in this direction, as it is difficult to validate values that capture the risk posed by systems that are still conceptual in nature. Additionally, unforeseen interactions and rare events, unknown operating conditions, environmental assumption violations, and errors introduced due to creeping system complexity may render traditional techniques of estimating risk for high confidence systems suspect. Additionally, the notion of AE may serve as a proxy for the difficulty in assuring a given function. For example, a system that possesses a single hazard and a single, independent (of other hazards and functions), and simple to implement mitigation for that hazard may be more efficacious than an approach that uses multiple mitigations with complex decision logic used to select amongst those mitigations. Factors such as the complexity (e.g., the number of components and the number of interactions between these components, etc.), uncertainty (i.e., epistemic and aleatoric), and architectural flexibility (e.g., tolerance to changing environmental assumptions, novel interactions, or evolving decision logic, etc.) associated with the mitigations may be indicative of whether the overall AE of the mitigation strategy with respect to the given hazard set may be high or low. Thus, there may exist a set of measures which are indicative of higher or lower AE when considered collectively. Clearly defining and validating such measures is a subject of future work as is determining how these measures inter-relate.

## VI. Conclusion

The concept of AE is seen as a parameter, criteria, or perspective by which to evaluate, identify and explore safety risk mitigation strategies and operational assurance architectures. Validation of the utility of this concept through flight testing is a first step in determining its potential role in assessing the overall safety of complex, increasingly autonomous systems that cannot be fully assured in the design phase.

## Acknowledgments

The authors would like to acknowledge the influence and input of Prof. Ella Atkins in fleshing out this concept.

## References

[1] Morgan Stanley, “eVTOL/Urban Air Mobility TAM Update: A Slow Take-Off, But Sky’s the Limit”, May 2021, https://assets.verticalmag.com/wp-content/uploads/2021/05/Morgan-Stanley-URBAN_20210506_0000.pdf

[2] E. Mazareanu, “Market size of the global airline industry 2018-2021”, August 18, 2021, Statista, https://www.statista.com/statistics/1110342/market-size-airline-industry-worldwide/

[3] Morgan Stanley, “Flying Cars: Investment Implications of Urban Air Mobility”, December 2018

[4] National Research Council. 2014. Autonomy Research for Civil Aviation: Toward a New Era of Flight. Washington, DC: The National Academies Press. https://doi.org/10.17226/18815.

[5] P Lael Rudd and Herb Hecht. Certification techniques for advanced flight critical systems. Technical report, WPAFB, 2008.

[6] C. Sánchez, G. Schneider, W. Ahrendt, et al. A survey of challenges for runtime verification from advanced application domains (beyond software). Form Methods Syst Des 54, 279–335 (2019). https://doi.org/10.1007/s10703-019-00337-w

[7] L. Sha, "Using simplicity to control complexity", IEEE Softw., no. 4, pp. 20-28, 2001.

[8] J. D. Schierman et al., Runtime assurance framework development for highly adaptive flight control systems, Charlottesville: Barron Associates, Inc., 2015.

[9] "Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions" in ASTM F3269, West Conshohocken PA: ASTM International, 2017.

[10] M. Clark, X. Koutsoukos, R. Kumar, I. Lee, G. Pappas, L. Pike, J. Porter, and O. Sokolsky, “A Study on Run Time Assurance for Complex Cyber Physical Systems,” AFRL Report, April 18. 2013. https://apps.dtic.mil/sti/pdfs/ADA585474.pdf

[11] A. Goodloe, “Challenges in High-Assurance Runtime Verification,” International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2016), Corfu, Greece, 2016. Springer, pp. 446-460

[12] A. Goodloe and L. Pike. Monitoring distributed real-time systems: A survey and future directions. Technical Report NASA/CR-2010-216724, NASA Langley Research Center, July 2010.

[13] N. G. Leveson. 1995. Safeware: system safety and computers. Association for Computing Machinery, New York, NY, USA.

[14] Federal Aviation Administration, “AC 23.1309-1E - System Safety Analysis and Assessment for Part 23 Airplanes”, FAA Advisory Circular, Nov 17, 2011.
https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1019681

[15] SAE ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, December 1, 1996.

[16] General Operating and Flight Rules, 14 Code of Federal Regulations § 91, January 2012.

[17] Operating Requirements, 14 Code of Federal Regulations § 135, January 2011.

[18] Small Unmanned Aircraft Systems, 14 Code of Federal Regulations § 107, January 2017.

[19] Federal Aviation Administration, Operation of Small Unmanned Aircraft Systems Over People, May 16, 2021
https://www.federalregister.gov/documents/2021/01/15/2020-28947/operation-of-small-unmanned-aircraft-systems-over-people

[20] S. Young et. al., “Architecture and Information Requirements to Assess and Predict Flight Safety Risks During Highly Autonomous Urban Flight Operations,” NASA/TM-2020-220440, January 2020.

[21] E. Dill, K. Hayhurst, S. Young, and A. Narkawicz, "UAS Hazard Mitigation through Assured Compliance with Conformance Criteria," AIAA SciTech Forum, Kissimmee, FL, Jan 8-12, 2018.
[22] E. Dill, R, Gilabert, and S. Young, "Safeguard – Flight Test Results of an On-board System Designed to Assure Conformance to Geospatial Limitations," 37th IEEE/AIAA Digital Avionics Systems Conference, London, UK, September 23-27, 2018.
[23] J. Kim, P. Sharma, E. Atkins, N. Neogi, E. Dill, and S. Young, "Assured Contingency Landing Management for Advanced Air Mobility," 40th IEEE/AIAA Digital Avionics Systems Conference, San Antonio, TX, October 4-8, 2021.
[24] A. Kotoky and C. Yap, "A Shortage of Pilots Looms as the Next Challenge for Airlines", Bloomberg, September 21, 2021, https://www.bloomberg.com/news/articles/2021-09-21/a-shortage-of-pilots-looms-as-the-next-challenge-for-airlines
[25] S. Young, E. Ancel, E. Dill, A. Moore, C. Quach, K. Smalling, K. Ellis, "Flight Testing of In-Time Safety Assurance Technologies for UAS Operations", AIAA Science and Technology", AIAA AVIATION Conference, Chicago, IL, June 2022.
[26] E. Ancel, S. Young, C. Quach, R. Haq, K. Darafsheh, K. Smalling, S. Vazques, E. Dill, R. Condotta, B. Ethridge, L Teska, T. Johnson," Design and Testing of an Approach to Automated In-Flight Safety Risk Management for sUAS Operations", AIAA AVIATION Conference, Chicago, IL, June 2022.
[27] National Aeronautics and Space Administration, "NASA Software Engineering Requirements (NPR 7150.2D)," URL: https://nodis3.gsfc.nasa.gov/displayDir.cfm?t=NPR&c=7150&s=2B, 2022, [retrieved on 12 March 2022].
[28] Federal Aviation Administration, "Package Delivery by Drone (Part 135)", URL: https://www.faa.gov/uas/advanced_operations/package_delivery_drone/, [retrieved on April 8, 2022].