ASTM F3269 - An Industry Standard on Run Time Assurance for Aircraft Systems

Pranav Nagarajan[1]
Technical University of Munich, Garching, 85748, Germany

Suresh K. Kannan, Ph.D.[2]
Nodein Autonomy, Burlington, Connecticut, 06013, USA

Mike E. Vukas[3]
Federal Aviation Administration, Kansas City, Missouri, 64106, USA

Christoph Torens[4]
DLR - German Aerospace Center, Braunschweig, 38108, Germany

George F. (Rick) Wilber[5]
Boeing, Seattle, Washington, 98124, USA

This paper discusses the philosophy and editorial considerations behind the ongoing second revision of the ASTM F38 Committee standard on run time assurance for aircraft systems – ASTM F3269, titled "*Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions*". It describes the key aspects of the Run Time Assurance (RTA) architecture as depicted in the current revision of the standard and provides some insights on the design best practices suggested in the standard. RTA is a certification strategy for unmanned aircraft systems that contain complex functions, which may not be certifiable using traditional design assurance practices. This challenge may arise in part due to the inherent algorithmic complexity of these functions. It may also be due to the inability to produce design assurance artifacts according to industry standards such as RTCA DO-178C (software) or DO-254 (hardware) for commercial off-the-shelf components used on-board the aircraft. RTA adds value not only to unmanned applications, but also to manned aviation – particularly in General Aviation (GA) and Advanced Air Mobility (AAM). It has the potential to enable technologies for autonomous aircraft systems and simplified vehicle operations. The strategy will also play a role in the design assurance and certification of adaptive controllers and functions using artificial intelligence and machine learning algorithms.

Abbreviations

A(/U)AM	Advanced (/Urban) Air Mobility	**Auto-GCAS**	Automatic Ground Collision Avoidance System
CF	Complex Function	**GA**	General Aviation

[1] Research Associate, Institute of Flight System Dynamics, AIAA Student Member.
[2] Chief Executive Officer, AIAA Member.
[3] Software Specialist, Policy and Innovation Division (AIR-694).
[4] Research Scientist, Institute of Flight Systems, Department Unmanned Aircraft, AIAA Senior Member.
[5] Technical Fellow, Chair of ASTM Working Group WK65056

LRU	Line Replaceable Unit	**RF**	Recovery Function
RS	RTA Switch	**RTA**	Run Time Assurance
SM	Safety Monitor	**SMTT**	Safety Monitor Trigger Threshold
SORA	Specific Operational Risk Assessment	**UAS**	Unmanned Aircraft Systems

I. Introduction

Certifying software systems containing algorithms that can misbehave is challenging. Difficulties arise when there is an unavailability of artifacts that assure that a given software implementation will work as expected every time to the level of confidence required for certification. In many cases, guarantees of "expected" behavior cannot be provided due to the inherent mathematical complexity, nondeterminism, or merely the monetary cost of arriving at such guarantees. In these cases, it is useful to detect misbehavior, reset, and switch to an alternate algorithm or, in some manner, bound the outputs of the algorithm so that a particular variable remains within pre-defined bounds. The implicit goal is to ensure that the overall system remains in a safe state. This is run time assurance. The ASTM standard F3269-17, published by the ASTM Committee on UAS (F38) in 2017, intends to offer some guidance to applying this strategy of RTA to enable the certification of unmanned aircraft systems. Based on feedback received from industry and certification authorities, a significant amount of effort has been directed towards revising and improving the first version of the standard.

This paper intends to provide some insights into that editorial effort within the responsible working group WK65056, the general philosophy behind the new version of the standard and considerations made in the new architecture presented in the second edition. The current revision is undergoing the ASTM balloting process at the F38 subcommittee and committee levels. While this paper attempts to give an overview of all aspects of the standard, the feedback received during the ongoing balloting process drives the focus of the paper on certain topics to a considerable extent. In this paper, the designation “F3269-17” is used when referring to the first and currently published version of the standard, whereas “F3269” without the year of publication is used as a moniker for the ongoing revision or as yet to be published second edition of the standard.

The contents of the paper are structured as follows. Following this introductory section, Section II provides a background on the proposal of run time assurance as a certification strategy and shows that the concept is discussed and demonstrated to a considerable extent in previous research. It also provides some context for the development of the first and now the second (in progress) edition of the standard. Section III provides an overview of the second edition of the standard itself in several sub-sections. Sub-section III-A discusses the concept of a so-called *Larger System* and its relation to operational safety. Sub-section III-B describes the generic RTA architecture depicted in the standard. It describes the various components of the RTA system and their associated attributes. Sub-section III-C discusses the concept of *RTA System Coverage* and the link it provides between the design assurance and run time assurance frameworks. This section also informs the reader on considerations made in specifying the pre-defined bounds for safe operation and the conditions for a switch from the complex function to the recovery function. Sub-section III-D discusses the integration of multiple monitors and recovery functions at different levels of the control loop in the RTA architecture. Sub-section III-E discusses the architectural variants represented by the various appendices in the standard. Finally, Sub-section III-F describes the goals and benefits provided by the standard from a certification perspective. Section IV summarizes the conclusions of this publication and identifies areas for future work based on own insights or feedback received from the broader community.

II. Background

This detailed overview of existing literature is intended to highlight the context in which the concept of RTA as a certification strategy has evolved over the previous decades. It is the hope of the authors, that understanding some of the excellent contributions to the field will also help comprehend the editorial decisions made during the development of the current revision of the standard.

Seto et al in 1998 [1] already proposed the term *simplex architecture* to describe the conceptual framework of RTA for a control system and Sha used it in 2001 [2] in an aerospace context. The simplex architecture in the latter publication proposes to use a *high-assurance control subsystem (in F3269: Recovery Function)* to protect a *high-performance control subsystem (in F3269: Complex Function)*. This, according to Sha, is the idea of using "simplicity to control complexity". The application of this architecture is limited in discussion to control systems. The idea of using this approach to not only assure operational safety but also to increase operational performance even for non-

safety critical systems is mentioned in the context of industrial control applications, such as semi-conductor manufacturing.

Clark et al applied the term *run time assurance* to this conceptual framework – especially in the context of cyber physical systems of which avionics systems are a type – in their work at the Air Force Research Laboratory (AFRL) in 2013 [3]. In their study, they discuss the state of the art in RTA boundary methods, for runtime monitoring and switching, and for model-based design for RTA. Clark et al's study provides an excellent mathematical basis for the computation of safe/unsafe boundaries, and the interested reader is encouraged to purview this work for further information on the topic. They even provide useful insights on which tasks should be considered in offline (*design-time*) computations and which tasks are to be performed online or in runtime computations. The latter could serve as the basis for future requirements for the Safety Monitor, RTA Switch and Recovery Function blocks of the RTA architecture presented in Section III-B. However, it should be noted that, the ongoing revision of the standard focusses on the elicitation of higher-level requirements for the above-mentioned blocks, with their implementation considered to be out of the scope of the task group's activities.

The seminal work, in our opinion, on the development of the RTA framework in the context of adaptive flight control systems is the study produced by Schierman et al in 2015 [4]. This work is the first, to our knowledge, to mention the technical challenges in the implementation of multi-level RTA systems for highly complex safety-critical systems. The study considers RTA architectures for "inner-loop control, outer-loop guidance, *ownship* flight management, and fleet mission planning elements". Within a single *RTA System*, the study further considers the possibility of deploying multiple *Recovery Functions*, which is an expansion of the *simplex architecture* proposed by Sha. Schierman et al also consider the certification challenges associated with RTA and construct a safety case argument for the certification of the *RTA System*. The approach presented in this study is especially important for existing systems, which need to be "retrofitted" with a "plug-and-play" RTA architecture to accommodate new *Complex Functions*. Furthermore, the discussion in the study on the need for and integration of certified collision avoidance systems may provide insights to comprehending the example on Auto-GCAS provided in an appendix (Appendix A) to the second edition of F3269.

The work of Hook et al in 2016 [5] on certification strategies using RTA for Part 23 (normal category aeroplanes) autopilot systems represents an eminent contribution to the field – especially in the context of the applicability of RTA to manned systems. They present the case for the increased automation of Part 23 aircraft, highlighting the potential gains to operational safety in the GA sector. Hook et al discuss the possible certification strategies for integrating "low cost, off the shelf, advanced autopilots" in existing aircraft by leveraging the RTA framework. The authors cross-reference results from the work of Schierman et al (discussed above) and consider them in the context of the GA autopilot problem. Finally, they provide a high-level assurance case for a simple RTA-based autopilot system and preliminary results from experimentation on manned and unmanned platforms. A key insight derived from this paper is that, the *Complex Function* need not be treated completely as a *black box*. This allows partial credit for any development assurance already performed on the *Complex Function*.

In 2017, the first edition of ASTM standard F3269 (F3269-17) captured the core tenets of RTA in the form of an architecture standard practice [6]. This work was carried out by the working group WK53403 of ASTM F38, the Committee on Unmanned Aircraft Systems. Similar to this very publication, the chair of WK53403 (Cook) produced a paper in 2017 [7] to discuss the philosophy behind the standard, its development and the generic RTA architecture. The interested reader is referred to this paper for its excellent insights on the first (and current) edition of the standard.

In 2018, Hook et al published a study on the initial considerations for a multi-layered RTA approach for UAS [8]. The authors propose a classification of the basic flight functions and associated subtasks to be automated for UAS operations. The results presented in this paper were based on research conducted in the “Traveler” Project of NASA Armstrong Flight Research Center. In this project, a so-called Expandable Variable Autonomy Architecture (EVAA) was developed which provoked discussions on modularity of functions, recertification after changes to the *complex function* and arbitration between multiple monitors in an RTA system. The work was further developed in another publication by Skoog, Hook and Ryan in 2020 [9], where they provide insights on leveraging ASTM F3269-17 for providing safe operations of a highly autonomous aircraft. In this paper, they also propose an initial risk-based scheme for arbitrating between different monitors and associated conflicts in a multiple-monitor architecture. Some of these ideas presented in the latter publication have already influenced the requirements on architectures with multiple monitors and multiple recovery functions in the ongoing second edition of the standard as discussed in this publication.

III. ASTM F3269

The following section provides an overview of the editorial considerations in the ongoing revision of ASTM F3269 within the Working Group WK65056 of the ASTM Committee F38. The terms of reference defining the scope for WK65056 may be summarized into four key objectives:

- Provide additional guidance on Safety Monitor design best practices
- Provide additional use cases as Appendices
- Provide additional information contrasting the F3269 approach with other architectural approaches
- Modify requirements to performance based to allow multiple implementation and implementation architectures

With these four objectives in mind, the following subsections delve deeper into some of the key considerations of the standard.

A. Safety of the *Larger System*

Based on the background provided in the previous section, it is easy to misconstrue run time assurance as a means of certifying complex flight controllers for UAS and/or manned applications. In reality, the approach ensures the safety of the operation by bounding a complex function using a certified RTA System. Moreover, RTA has potential far beyond control applications and could be used at the input, processing or control and output level. For example, RTA can be applied to a sensor fusion algorithm, a neural network flight controller or an actuation system without significant changes to the generic architecture presented in the next section. While the top-level system for which assurance is argued may be the aircraft [10], it may also simply be the propulsion system [11]. RTA may be used at the individual LRU or function or system level without necessarily affecting the trajectory of the aircraft [10]. Therefore, in order to allow the use of the standard for a broad range of applications, the overall system for which RTA is used, is intentionally called the *Larger System*. It is the discretion of the applicant to define this larger system for their particular use case in cooperation with the certification authority. The applicant may also choose to simply bound the behavior of this so-called *Larger System* and allow the complex function control within these bounds. This concept can support risk-based approaches to granting access to the airspace. The EASA specific category utilizes a so-called specific operation risk assessment (SORA), so that the requirements for development, maintenance and operation of the unmanned aircraft can be scaled individually to the mission scenario. A safe operation monitoring, enabled by the discussed RTA architecture, can mitigate risks of the operation [12]. Examples for operational limitations that can be bounded using RTA are the time or duration of the operation, cruise altitude, cruise speed, area of operation, or size and weight of the aircraft [13]. This approach allows for RTA to be implemented without directly monitoring the complex function by, instead, monitoring the *Larger System* behavior.

B. Generic RTA Architecture

This section describes the generic RTA architecture (depicted in Figure 1) as presented in the ongoing revision of the standard. The architecture and this publication contain some terms unique to the standard, which are described in Table 1.

Table 1 Terms described in the RTA architecture of ASTM F3269

Term	Description
Assured Data	information that may be directly used by RTA components and the Larger System
Assured Function	hardware and software items for which the UAS manufacturer produces sufficient evidence that these items function to the acceptable level of performance
Complex Function	a function that is performed by an untrusted, less trusted or unassured system
Input Manager	an assured RTA function that accepts assured and unassured data and conditions, validates and performs consistency checking, and outputs assured data to RTA components
Recovery Function	an assured RTA function that generates bounded RTA Output intended to keep the Larger System in a safe state
RTA Components	the set of assured functions defined in the RTA architecture, includes Input Manager, Safety Monitor, RTA Switch, Recovery Function(s)
RTA Switch	an assured RTA function that receives an input/data from the safety monitor and switches to the appropriate recovery function
RTA System	functional scope that is covered by the RTA and at a minimum contains RTA Components and a Complex Function

Term	Description
Safety Monitor	an assured RTA function which continuously evaluates Larger System and/or complex function behaviors, with the intent of discovering misbehavior of the complex function When necessary, the monitor selects and commands the RTA Switch to the selected recovery function.
Unassured Data	is information that is unassured and hence may not be directly used by RTA components, unless put through data conditioning and/or dynamic consistency checking
Unassured Function	any function that is not assured to acceptable certification standards

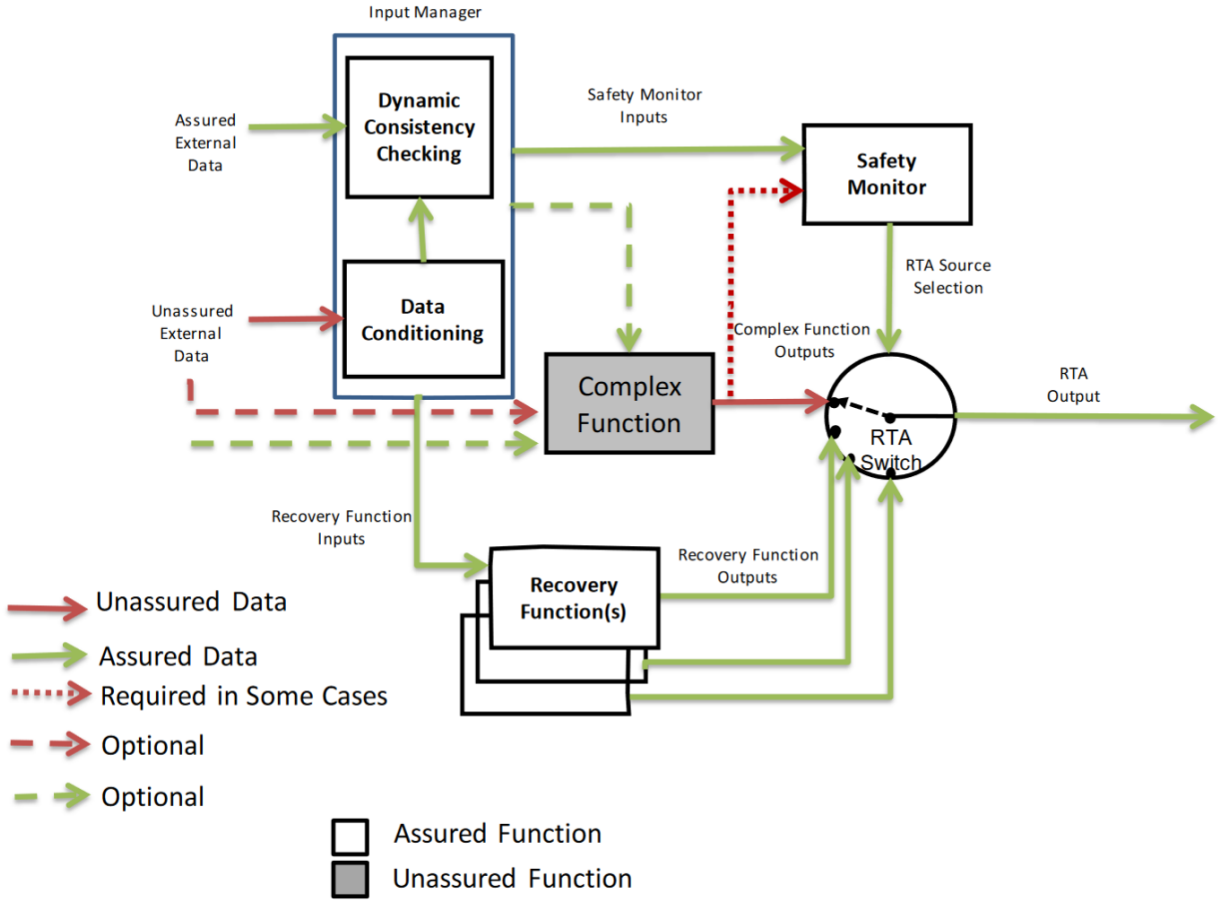


Figure 1 RTA architecture as presented in the ongoing revision of ASTM F3269

The generic architecture allows for two primary variants of the RTA framework to be implemented. In the first variant, the Safety Monitor observes the behavior of the Larger System and upon detecting a misbehavior of the Larger System, switches from the Complex Function to the Recovery Function. In the second variant, marked with additional dotted lines in Figure 1, the Safety Monitor may make this decision by monitoring the Complex Funtion directly. The dashed lines represent optional implementation choices which are not required for either variant of the architecture. While it is not explicitly the main purpose of the RTA framework, it is of course possible – and may even be desired – to allow also for switching between the Recovery Functions and a *graceful degradation* [14] of functional capabilities. The availability and integrity of the Recovery Functions should be considered during the decision-making process in the Safety Monitor. These aspects – while deliberated upon within the editorial committee – do not appear in the *requirements* for RTA Components, as such guidance was considered to be implementation-specific and prescriptive in nature. However, the standard provides some recommendations for each architecture component in separate *best practices* sections. The standard also refers to design best practices developed by ASTM's autonomy design and operations in aviation administrative committee – AC377 [15][16].

One key aspect of the RTA architecture that is worthy of further discussion is the routing of the Complex Function Output. As mentioned in Table 1, unassured data should not be used directly by the RTA Components. We see in

Figure 1, that the Complex Function Output is being consumed directly both by the RTA Switch and the Safety Monitor. It is currently being discussed within the Working Group, whether one or both of these signals should be routed through the Input Manager. This should not be interpreted to mean, however, that the Complex Function Output will be changed by the Input Manager. Rather, the proposed re-routing is intended to allow for the data conditioning and dynamic consistency checking to be performed on the unassured data output by the Complex Function. Moreover, while the Input Manager is depicted in this architecture as a separate entity, designers may choose to implement the functionality of this block in each of the remaining RTA Components themselves. At any case, the designer should ensure that the RTA Components in general, and the Safety Monitor in particular, are protected from corruption by any unassured data.

C. Assurance Level and Coverage of the RTA System

During development, the applicant should first evaluate the assurance level required for the functionality fulfilled by the complex function. The RTA components used to bound this complex function must have the same assurance level. For example, if the complex function were safety-critical with a required design assurance level of FDAL A, then the RTA components that bound this complex function would need FDAL A. While the recovery functions used to bound a misbehaving complex function may have lower performance, the recovery functions must safely perform throughout the scope of the complex function's operation.

The concept of RTA System Coverage is key to understanding the effectiveness of the RTA framework. To define the overall RTA System Coverage, one begins by defining the *complex function's intended operational scope* and expresses it in dimensions the designer deems appropriate. Examples include speed envelope, control authority, field-of-view, or sensor range. This intended scope is then investigated based on currently available monitors and recovery functions or ones that can be developed. For each operating point of the complex function, two conditions must be true:

a) A monitor capable of detecting complex function misbehavior and triggering a suitable recovery function must be available.

b) A recovery function that ensures the larger system can be kept in a safe state must be available.

Any gaps in the combined (monitors, recovery function) coverage must be addressed by developing new monitors and recovery functions or disallowing operation at these points (Figure 2). At the end of the process described in Figure 3, the intended operational scope of the Complex Function is limited to that enabled by RTA System Coverage, i.e. all those points where at least one monitor function and one recovery function are available.

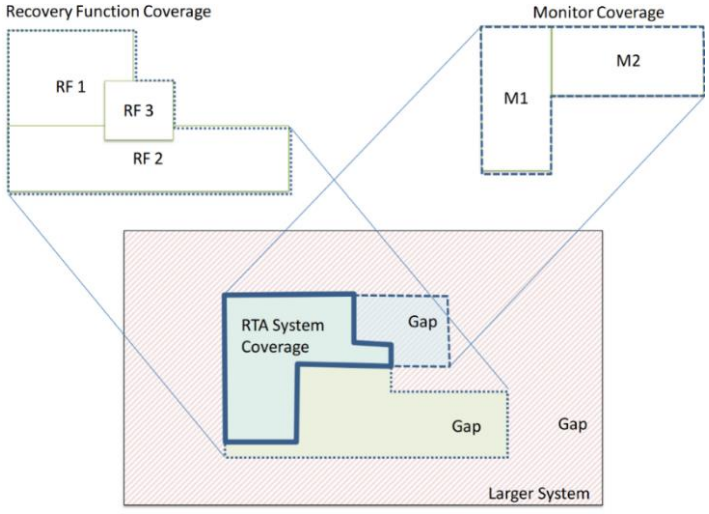


Figure 2 RTA System Coverage

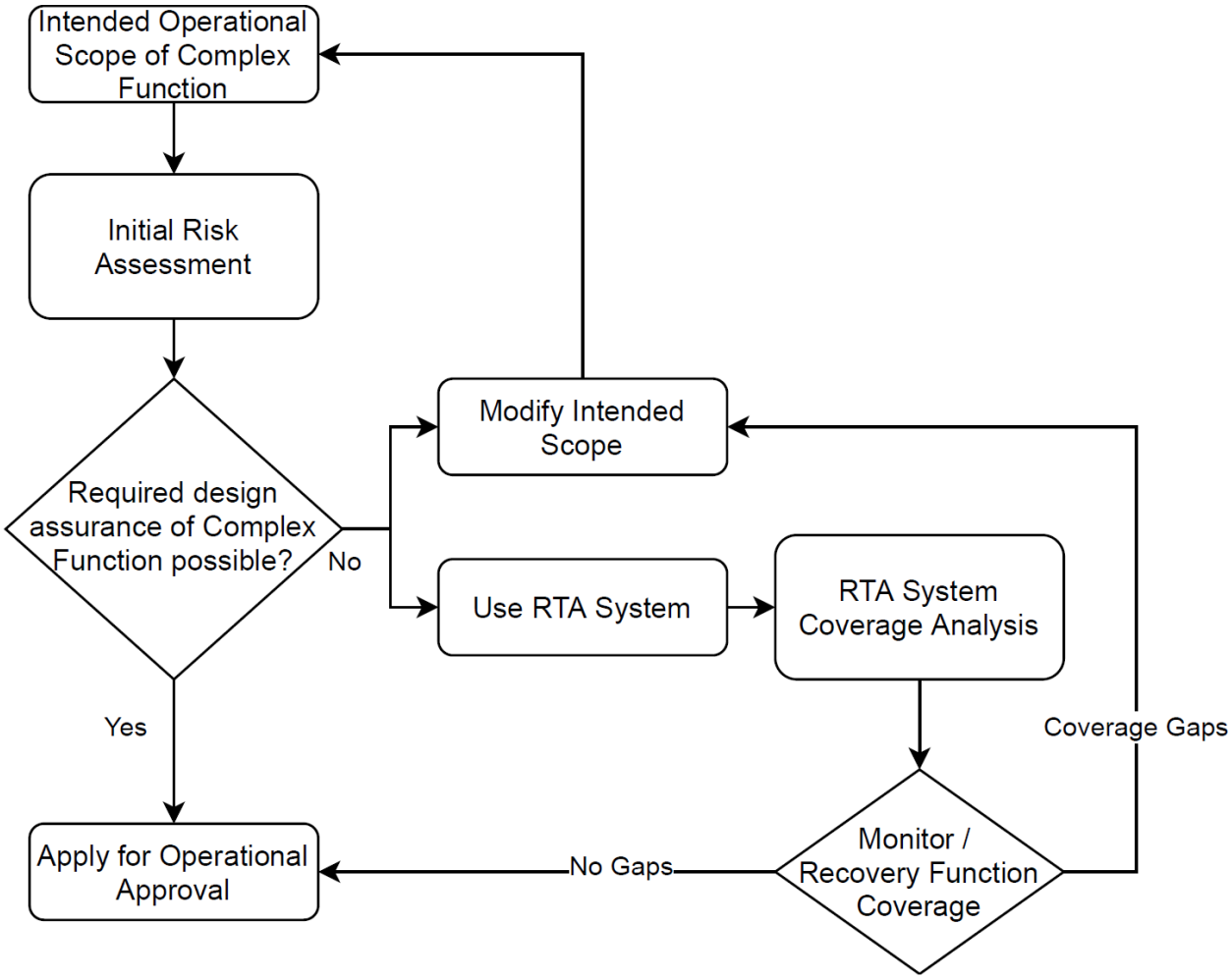
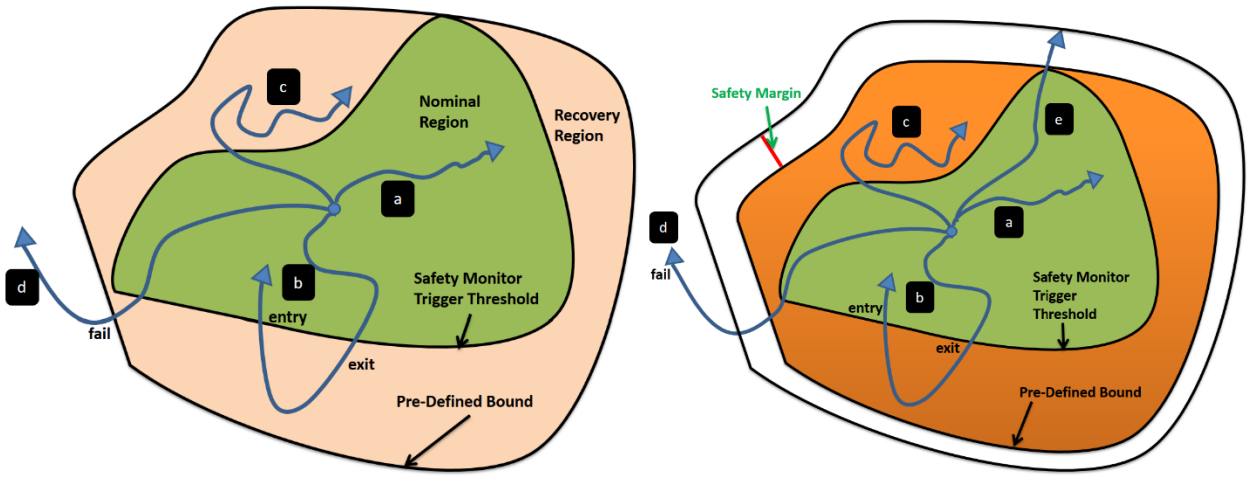


Figure 3 Process to arrive at operational scope of Complex Function



(a) Operating modes of the RTA system (b) Safety margins

Figure 4 Defining operational bounds for the RTA system

Figure 4(a) illustrates the various modes the RTA system can find itself in. The system is designed to operate in the nominal region under the control of the complex function. When the larger system or complex function exceeds the safety monitor trigger threshold, the safety monitor detects this and switches to a suitable recovery function. It may

be desirable to return control to the complex function if the system re-enters the nominal region. Engineering considerations such as stable switching and chattering must be addressed to ensure safe operation and avoid frequent switching between the complex and recovery functions. In defining the safety monitor trigger threshold, consideration has to be given to the ability of the recovery function to bring the system back to or keep it in a safe state. Furthermore, the operation of the complex or recovery function may only be safe in a subset of the operational domain of the Larger System. It is up to the applicant to allow for sufficient safety margins in determining their pre-defined bounds for the operation of the RTA system, as depicted in Figure 4(b). The diagram with safety margins is part of work done during the risk assessment process and is not depicted in the standard, as this is out of scope for the purposes of the standard. Finally, considerations of timing and latency play an essential role in determining the capability and implementation of the RTA system.

D. Multi-level RTA architectures

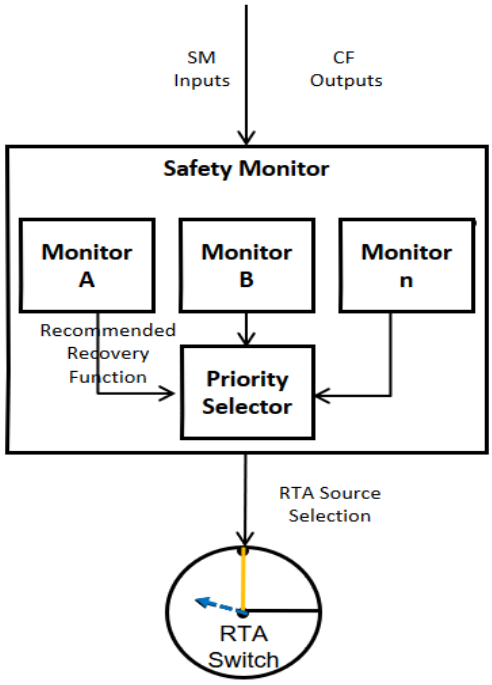


Figure 5 Depiction of an example multi-monitor architecture

One of the newer additions to the second addition of ASTM standard F3269 is a multi-monitor architecture. The Safety Monitor may be comprised of several Monitor Sub-Functions, which may provide coverage at different operating points. Each Monitor Sub-Function may recommend an appropriate Recovery Function, which depending on the architecture may be identical or different Recovery Functions. This allows for the architecture to be used simply for voting among different monitors or to choose different recovery actions for different operational scenarios. This aspect of the standard was also influenced, as discussed in Section II, by the seminal work of Schierman et al in 2015, the work of Hook et al in 2018 and the more recent work of Skoog, Hook and Ryan in 2020. A key requirement is that on the Safety Monitor for a unique RTA Source Selection, so that the RTA Switch receives an unambiguous input to select the appropriate RTA Output. It was decided that any arbitration to decide the nature of the RTA Output (Complex or Recovery Function Output) should be performed within the Safety Monitor. Other designers may indeed place this arbitration, depending on the preference of implementation, in the RTA Switch. While there is a requirement imposed on the designer to provide a clear priority selection logic during the design of the multi-monitor architecture, the implementation is not prescribed. Furthermore, the RTA architecture is also designed to be composable. If a given functionality, F, can be decomposed into A, B, and C, each subsystem A, B, C may be run-time-assured independently. Moreover, the overall functionality F can be run-time-assured independently.

E. Architectural variants in the appendices of ASTM F3269

One of the objectives defined in the terms of reference for WK65056 was to include different architectural variants of RTA as examples in the appendices of the new edition of the standard. Experts from regulatory bodies, academia and industry contributed to these examples. Three of these examples are described in this publication, as they describe widely varying applications of the same architecture framework.

The first example was provided by Mike Vukas, a Software Specialist in the Policy and Innovation Division of the Federal Aviation Administration. The example system shown here uses a Machine Learning AI Autopilot (MLAA) that has the capability to learn adaptively/online improving its flying performance while providing envelope protection, thus always keeping the pilot in a safe state. If the MLAA tries to perform an unsafe maneuver, the RTA must intervene to meet the safety requirements. The architecture shown in Figure 6 incorporates a multi-layer protection scheme, which not only provides a Recovery Function for a failure of the MLAA, but also uses the pilot as a secondary backup Recovery Function. The pilot can take control of the aircraft by either disengaging the autopilot function or by directly taking command of the yoke.

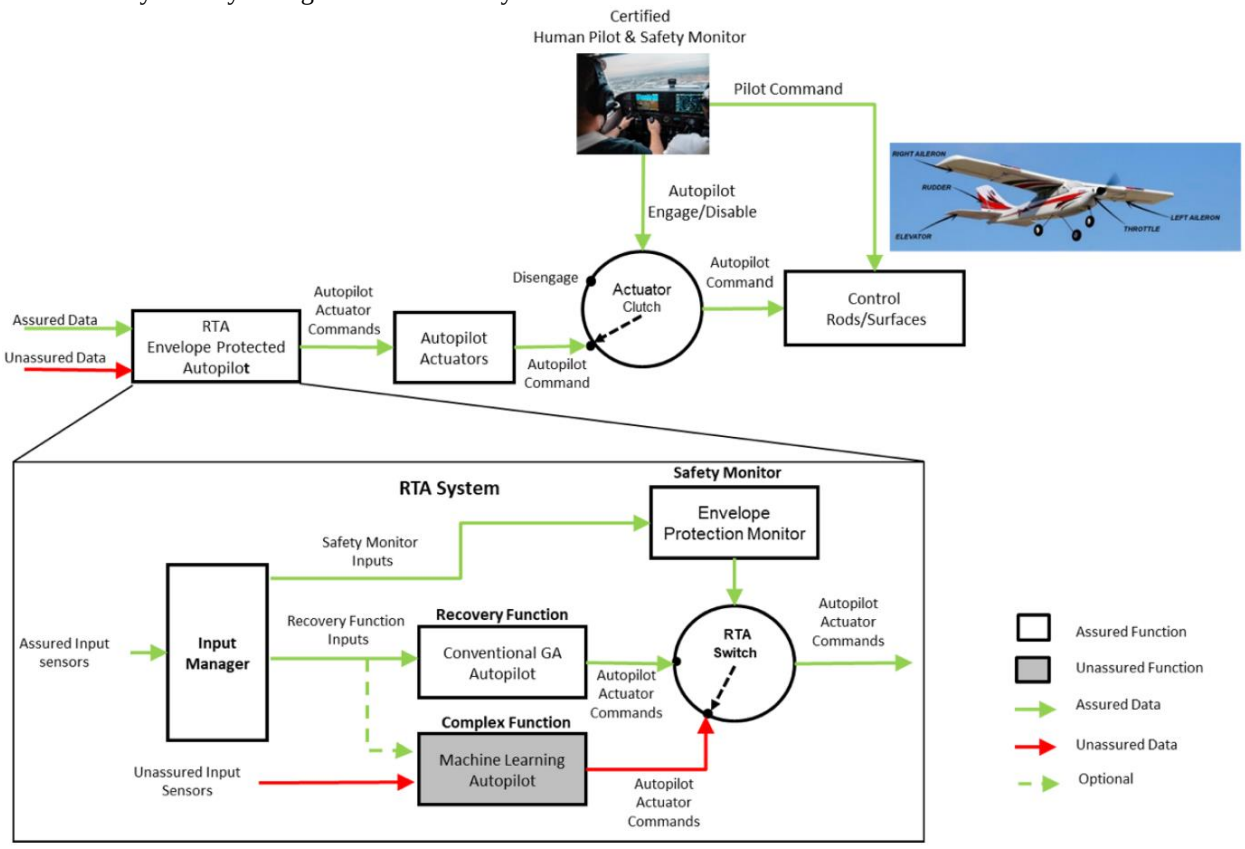


Figure 6 RTA system for a machine learning adaptive autopilot retrofitted using RTA

The MLAA RTA architectural design shows how a two-tier backup safety approach can be engineered into systems today using a fail-safe architectural approach; the first is the bounding of the MLAA using Run-Time Assurance and the second being the pilot disengaging mechanism to allow manual control. A resilient architecture can be used to assure safety enhancing systems that would otherwise never be allowed on an aircraft due to the certification challenges and costs. This approach can ultimately lead to certifying new and novel systems using existing design assurance practices while relying on fail-safe system architectures that increases the safety in the GA market.

The second example was provided by Dr Suresh Kannan at Nodein Autonomy. Figure 7 illustrates the simplified functional architecture used to fly an experimental aircraft over the past two decades. The use of a human safety pilot to serve as a catch-all recovery function when adverse behavior is detected is a prevailing paradigm used by autonomous vehicle flight test programs. Early flight tests involved development and testing of the core adaptive flight control system that tracks all 6 degrees of freedom of the aircraft using the actuation vector $\boldsymbol{\delta}$. The controller uses a multi-layer neural network to learn the differences between the aircraft model used to design the controller and the aircraft's actual dynamics.

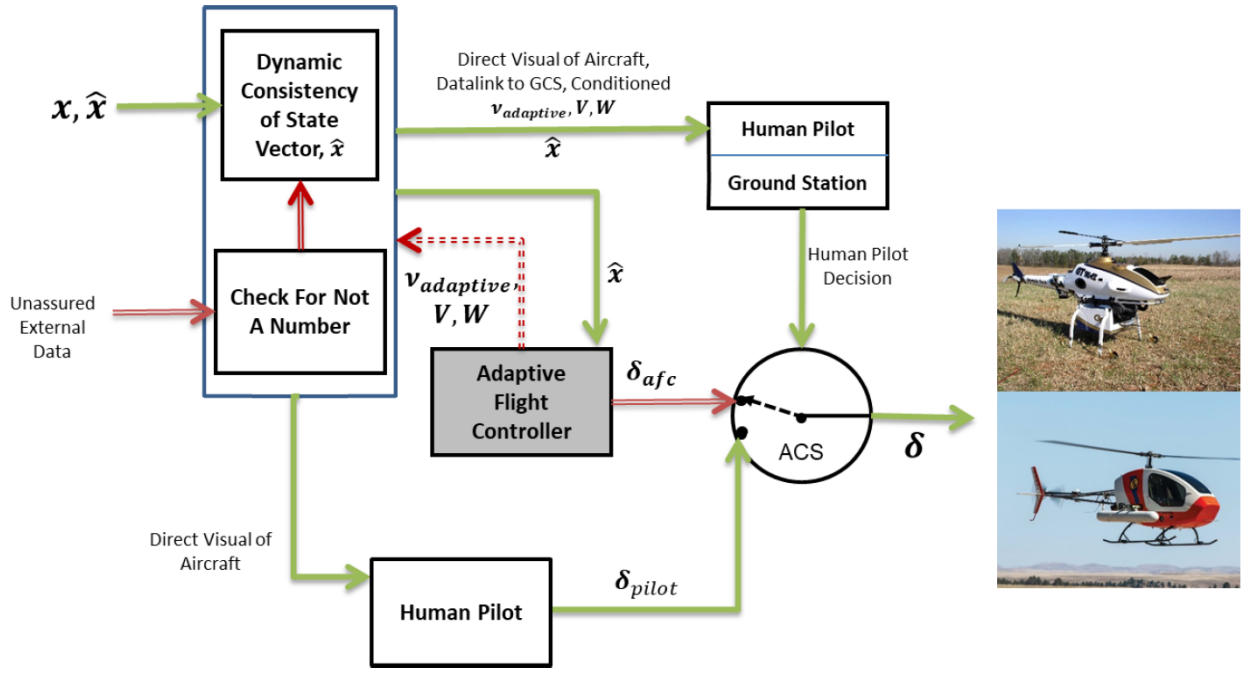


Figure 7 RTA system for an adaptive controller with human-in-the-loop

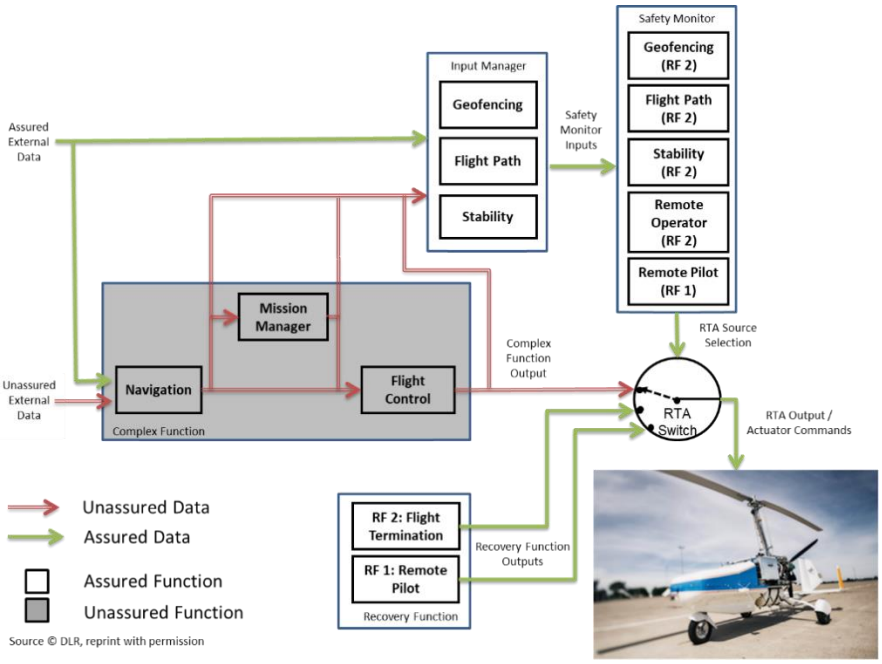


Figure 8 ALAADy, an implementation of RTA by the German Aersopace Center

The third and final example from the standard discussed in this paper – on Automated Low Altitude Air Delivery (ALAADy) – was provided by Christoph Torens at the German Aerospace Center. ALAADy represents a feasibility study for air cargo delivery using the SORA risk assessment, mentioned in section III-A. The risk assessment of a small aircraft flying over unpopulated areas will result in a low risk, compared to the risk that is assumed for certified aircraft. The RTA architecture is used to implement a geofence functionality. In case the geofence is violated, the RTA will trigger a termination of the flight, ensure that the aircraft does not fly over a populated area. This supports this safety of the operation. A simplified view on the ALAADy system architecture is depicted in Figure 8.

F. Certification aspects

Presently, Civil Aviation Authorities (CAA) require a traditional path to certifying the design of avionics systems that begin with the assumption that, each software and hardware component installed on an aircraft contributes independently to the safe operation of the aircraft. At the core of this process is an assessment of the risks associated with the functional failure that each system, assembly, component and software to ensure that the aircraft meets the required safety objectives. This is known as design-time assurance. RTA presents a framework for mitigating the risks associated with an unassured function that differs from the traditional certification paradigm that allows the implementation and installation of unassured functions that would otherwise never be accepted due to certification challenges and costs. ASTM F3269 provides an architectural framework for developing an RTA system as an alternative to design-time assurance to fulfill safety requirements of an unassured or complex function.

The following benefits of using run-time assurance apply from a certification perspective:

- The ability to mitigate hazards related to potential non-deterministic or unexpected behavior from unassured functions that employ advanced software methods or algorithmic complexity that cannot be certified using traditional certification practices.
- The ability to use functions that do not adequately provide the required artifacts for conventional DO-178 or DO-254 assurance practices.
- The ability to use COTS hardware and/or software for the unassured function.
    - For example automotive components, thereby leveraging mature software with extensive service history that was developed for other safety critical industries, but cannot be shown to comply with aviation development assurance practices.
    - For example industry components where source code or other associated engineering artifacts are unavailable.
- Reduction in cost and schedule burdens typical of the higher assurance levels of the assured RTA system, allowing rapid design iterations of the complex function after initial certification. A stated goal of the version of the standard is to allow unassured function upgrades after initial certification to minimize modifications to the certification.

IV. Conclusions and Outlook

This paper provided an overview of the editorial considerations behind the ongoing revision of the ASTM standard F3269. While the standard itself is undergoing the ASTM balloting process currently, the core concepts in the standard representing the major changes from the first (and current) edition of the standard – ASTM F3269-17 – were discussed. The generic architecture was presented and the different implementation possibilities were showcased using examples contributed to the standard by co-authors of this paper. The importance of operational safety was highlighted in the definition of the Larger System and the discussion on RTA System Coverage respectively. A brief overview of multi-monitor architectures was given and the possibility of designing composable RTA architectures was noted. The standard in its final version, when published, may present some deviations from illustrations in this paper. Future work – whether in further research or revisions of the standard itself (both within the F38/UAS and F44/GA committees in ASTM) – should consider the added value RTA may bring to operational safety in more detail than dealt with in the current revision of F3269. Moreover, special considerations in the design of the RTA framework for piloted operations (and the inclusion of such information in F3269 or in separate guidance material) could be deliberated upon in a joint effort across the ASTM F38 and F44 committees. Finally, researchers should consider the problem of arbitration between multiple monitors and multiple recovery functions in greater detail.

Acknowledgments

The authors would like to acknowledge the support of all participating members of the working group WK 65056 for their inputs to the development of the current revision of F3269. In particular, we would like to thank – Ryan Spoelhof and Gary Goz at GE Aviation; Peter Lyons at Overair; José Martin at Aurora Flight Sciences; Robert Jones, Ritesh Ghimire and Rick May at the Federal Aviation Administration; Prof. Loyd Hook at the University of Tulsa; Dr. Kerianne Hobbs and Dr. John Schierman at the Air Force Research Laboratory. We would like to further acknowledge Dr. Stephen Cook (chair of former Working Group WK53403, responsible for the F3269-17) at Northrop Grumman for his guidance and suggestions for improvement to the current revision. We would like to thank Anna

Dietrich at Anna Dietrich Consulting/Xwing for her efforts in coordinating with ASTM F44.50 (General Aviation, Systems Committee) to gather their inputs to the standard and the members of F44.50 for participating in the current balloting process. We would like to offer our special thanks to Mary Mikolajewski at ASTM for her continuing administrative support.

References

1. Seto, D., Krogh, B., Sha, L., Chutinan, A., "The Simplex Architecture for Safe Online Control System Upgrades," Proc. American Control Conference. Philadelphia, PA, June, 1998, pp. 3504-3508.
2. Sha, Lui. "Using simplicity to control complexity." *IEEE Software* 4 (2001): 20-28.
3. Clark, Matthew, et al. *A study on run time assurance for complex cyber physical systems*. Air Force Research Lab Wright-Patterson AFB OH, Aerospace Systems Directorate, 2013.
4. Schierman, John D., et al. *Runtime assurance framework development for highly adaptive flight control systems*. No. AFRL-RQ-WP-TR-2016-0001. Barron Associates, Inc. Charlottesville, 2015.
5. Hook, Loyd R., et al. "Certification strategies using run-time safety assurance for part 23 autopilot systems." *2016 IEEE Aerospace Conference*. IEEE, 2016.
6. ASTM International. *F3269-17 Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions*. West Conshohocken, PA; ASTM International, 2017. doi: https://doi.org/10.1520/F3269-17
7. Cook, Stephen P. "An ASTM Standard for Bounding Behavior of Adaptive Algorithms for Unmanned Aircraft Operations." *AIAA Information Systems-AIAA Infotech@ Aerospace*. 2017. 0881.
8. Hook, Loyd R., et al. "Initial considerations of a multi-layered run time assurance approach to enable unpiloted aircraft." *2018 IEEE Aerospace Conference*. IEEE, 2018.
9. Skoog, Mark A., Loyd R. Hook, and Wes Ryan. "Leveraging astm industry standard f3269-17 for providing safe operations of a highly autonomous aircraft." *2020 IEEE Aerospace Conference*. IEEE, 2020.
10. Benders, Sebastian, et al. "Softwarearchitektur für einen unbemannten Luftfrachttransportdemonstrator." *Deutscher Luft-und Raumfahrtkongress*. 2018.
11. Wong, Edmond, et al. "Towards Run-time Assurance of Advanced Propulsion Algorithms." *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. 2014.
12. Torens, C., Nikodem, F., Dauer, J.C. et al. Geofencing requirements for onboard safe operation monitoring. CEAS Aeronaut J 11, 767–779 (2020). https://doi.org/10.1007/s13272-020-00451-0
13. Joint Authorities of Rulemaking for Unmanned Systems, "JARUS guidelines on Specific Operations Risk Assessment (SORA) V2.0," 2019.
14. Shelton, C. P., Koopman, P., and Nace, W., "A framework for scalable analysis and design of system-wide graceful degradation indistributed embedded systems,"Proceedings of the Eighth International Workshop on Object-Oriented Real-Time DependableSystems, 2003. (WORDS 2003), IEEE, Guadalajara, Mexico, 15-17 Jan. 2003, pp. 156–163. doi:10.1109/WORDS.2003
15. ASTM International, Developmental Pillars of Increased Autonomy for Aircraft Systems, West Conshohocken PA, ASTM International, 2020.
16. S. Cook, A. Dietrich, L. Hook, W. Ryan and D. M. Stevens, "Advancing Autonomy in Aviation: A Holistic Approach," 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 2020, pp. 1-8, doi: 10.1109/DASC50938.2020.9256568.