Designation: ~~F3269 – 17~~ F3269 – 21

Standard Practice for Methods to Safely Bound ~~Flight~~ Behavior of ~~Unmanned~~ Aircraft Systems Containing Complex Functions Using Run-Time Assurance[1]

This standard is issued under the fixed designation F3269; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

INTRODUCTION

This practice defines an architecture using Run-Time Assurance (RTA) in conjunction with unassured functions or commercial off-the-shelf (COTS) functions that have not been developed to traditional aerospace standards and processes. This section provides the scope, applicability, and intended use for the understanding of this practice.

The practice is organized as follows: *(1)* An introduction, background, and scope to provide context for applying the capabilities defined in this practice to unmanned aircraft system (UAS) certification, or operational approval, or both. *(2)* Definitions of key terms and abbreviations. *(3)* Description of a Run-Time Assurance (RTA) architecture. *(4)* Appendixes that contain Examples of RTA in systems and supplemental information. *(a)* Ground Collision Avoidance System (GCAS) as an Example RTA. *(b)* Machine Learning AI Autopilot (MLAA). *(c)* Run-Time Assurance for a Neural Network-Based Adaptive Flight Control of an Unmanned Aircraft. *(d)* Run-Time Assurance for Risk-Based Operation. *(e)* Example Implementation of Timing and Latency Requirement. *(5)* A list of documents referenced herein.

BACKGROUND

There is significant interest from industry and civil aviation authorities (CAA) to have a standard practice to enable new and novel technologies used in UAS operations containing unassured or COTS functions/systems, or both, to be used on certified aircraft and aviation systems. From this point forward, "functions/systems" will be referenced as "functions." Developing a certification path for these technologies may also introduce greater safety to aviation.

In this practice, the term *Complex Function* (CF) may be any function, algorithm, component, or system that has not been subject to accepted CAA or aerospace design assurance practices, or both (DO-178C, DO-254, ARP4754A, etc.). Motivations to use such an unassured function arise from the need or desire to use commercial, off-the-shelf systems or parts that have algorithmic complexity, probabilistic algorithms, fuzzy logic, environmental uncertainties, or no pedigree. The complexity may also come from factors associated with new and novel technologies such as sensor measurement precision, nondeterministic algorithms, data-driven algorithms, or artificial intelligence (for example, machine learning, genetic algorithms). A complex function may be any combination of software or hardware.

Traditional approaches to digital avionics design begin with the assumption that each software and hardware component on an aircraft contribute independently to the safe operation of the platform. At the core of this process is an assessment of the risks associated with the functional failure of each system, assembly, or component to ensure that the aircraft meets the required safety objectives. This is known as design-time assurance.

[1] This practice is under the jurisdiction of ASTM Committee F38 on Unmanned Aircraft Systems and is the direct responsibility of Subcommittee F38.01 on Airworthiness.
Current edition approved ~~Sept. 1, 2017~~July 15, 2021. Published ~~September 2017~~November 2021. Originally approved in 2017. Last previous edition approved in 2017 as F3269–17. DOI: ~~10.1520/F3269-17.~~10.1520/F3269-21.

This practice describes a run-time assurance method, which may be used as an alternative means to or in combination with design-time assurance. RTA mitigates the risk of complex function misbehavior by managing the system's use of the Complex Function output. The RTA includes a safety monitor, which monitors the complex function or the behavior the complex function has on the system, or both, at run-time. In the event the safety monitor determines that the complex function is not operating correctly, or is driving the system to an unsafe state, it disengages the complex function and initiates a recovery function.

This practice provides an RTA architecture and best practices that provide guidance to an applicant for ensuring that the behavior of an unmanned aircraft system (UAS) containing complex functions maintains the acceptable level of safety.

At the time of this practice's development, there is no accepted formal guidance material for certifying commercial UAS containing complex functions. Emerging CAA certification guidance, processes, and concepts have been considered in the development of this practice.

1. Scope

1.1 ~~This standard practice defines design and test best practices that if followed, would provide guidance to an applicant for providing evidence to the civil aviation authority (CAA) that the flight behavior of an unmanned aircraft system (UAS) containing complex function(s) is constrained through a run-time assurance (RTA) architecture to maintain an acceptable level of flight safety.~~The scope of this practice includes the following:

1.1.1 A set of components that comprise an RTA system.

1.1.2 Requirements and best practices to determine safe boundaries and RTA system coverage.

1.1.3 Requirements and best practices for an RTA system and RTA components, as applicable.

1.1.4 Appendixes with examples that demonstrate key RTA system concepts.

1.2 RTA components are required to meet the design assurance level dictated by a safety assessment process. Guidance for the safety assessment process may be found in references appropriate for the intended operations (ARP4754A, ARP4761, Practice F3178, etc.).

1.3 This practice ~~will have the benefit of enabling highly automated UAS operations. It is envisioned that applicants will use this practice as a means of compliance for safe implementation of complex functions for routine operations.~~was developed with UAS in mind. It may be applicable for aspects of manned aircraft certification/approval, as well as aviation ground systems. The scope of this practice is also envisioned to allow a variety of aircraft implementations where a human may perform the role of either the Complex Function or a Recovery Function.

1.4 ~~Verification of complex functions is considered too challenging to use conventional software assurance methods such as RTCA DO-178C or IEC 61508. Certification challenges under these standards include generating required artifacts, such as requirements, elimination of unintended functionality, traceability/coverage, and test cases required for verification.~~The scope of this practice does not cover aspects of hardware/software integration. These should be considered separately during the development process.

NOTE 1—This practice does not suggest a one-size-fits-all strategy knowing that not all use cases may fit well into this architecture. There may exist additional components required to satisfy specific applications to the practice.

1.5 ~~There is significant interest from industry and CAAs to have a standard practice to enable flight operations for UAS containing complex functions. Developing a certification path for these UAS technologies could also advance safety in General Aviation.~~The values stated in inch-pound units are to be regarded as standard. No other units of measurement are included in this standard.

1.6 ~~The following design tenets are offered to provide guidance to the UAS manufacturer as to the intended application of this standard.~~*Table of Contents:*

~~1.5.1 The RTA Architecture is intended to be used for Complex Functions that would require an amount of effort that is beyond reasonably practicable to pass CAA conventional certification requirements.~~

~~1.5.2 The UAS manufacturer should engage in appropriate design, test, and validation activities to enable the Complex Function to perform as intended.~~

~~1.5.3 The complexity of the Recovery Control Function (RCF) deterministic commands should be minimized insofar as practicable.~~

~~1.5.4 Repeated invocation of an RCF during a single mission may be considered an indication of improper Complex Function performance.~~

~~1.5.5 An RTA design with multiple RCFs should consider the aircraft state, relative outcomes, and differences in RTA recovery times in prioritizing the recovery actions in the safety monitor.~~

~~1.5.6 The UAS manufacturer should strive to minimize false or nuisance triggers of one or more RCFs as these false alarms undermine user confidence in the system and impact operational efficiency.~~

1.7 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety, health, and environmental practices and determine the applicability of regulatory limitations prior to use.*

1.8 *This international standard was developed in accordance with internationally recognized principles on standardization established in the Decision on Principles for the Development of International Standards, Guides and Recommendations issued by the World Trade Organization Technical Barriers to Trade (TBT) Committee.*

2. Referenced Documents

2.1 *ASTM Standards:*[2]

~~F3201~~F3060 ~~Practice for Ensuring Dependability of Software Used in Unmanned Aircraft Systems (UAS)~~Terminology for Aircraft

F3178 Practice for Operational Risk Assessment of Small Unmanned Aircraft Systems (sUAS)

F3341/F3341M Terminology for Unmanned Aircraft Systems

ASTM AC377 TR2-EB Developmental Pillars of Increased Autonomy for Aircraft Systems

2.2 *FAA Advisory Circular:*[3]

AC 23.1309-1E System Safety Analysis and Assessment for Part 23 Airplanes

2.3 ~~*Civil Standards, Policy, and Guidance:*~~*RTCA Standards:*

~~IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems~~~~[3]~~[4]

RTCA DO-178C Software Considerations in Airborne Systems and Equipment Certification

RTCA DO-254 Design Assurance Guidance for Airborne Electric Hardware

2.4 *SAE Standards:*[5]

SAE ARP4754A Guidelines for Development of Civil Aircraft and Systems

SAE ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment

3. Terminology

3.1 This section defines key terms and abbreviations for this practice.

3.2 Note on terminology: *shall versus should versus may*—use of the word "shall" implies that a procedure or statement is mandatory and must be followed to comply with this practice, "should" implies recommended, and "may" implies optional at the discretion of the supplier, manufacturer, or operator. Since "shall" statements are requirements, they include sufficient detail needed to define compliance (for example, threshold values, test methods, oversight, and references to acceptable industry standards). "Should" statements represent best practices to guide in the development of RTA Systems. "May" statements are provided to clarify acceptability of a specific item or practice and offer options for satisfying requirements.

3.3 *Unique and Common Terminology*—Terminology used in multiple standards is defined in F3341/F3341M, UAS Terminology Standard, and F3060, Aircraft Terminology Standard.

3.4 *Definitions of Terms Specific to This Standard:*

~~3.1.1 *complex function*—software function or algorithm that may cause the UAS to operate in a manner that is difficult to predict due to compounded implications from factors such as sensor measurement precision, algorithm complexity, environmental variables (for example, gusts, traffic, electromagnetic effects, etc.), multi-core processing, probabilistic algorithms, fuzzy logic, machine learning, genetic algorithms, resource availability, and aircraft system state. These software functions or algorithms are sometimes referred to as "autonomous", "non-deterministic", "artificial intelligence", "adaptive", or "intelligent" algorithms.~~

[2] For referenced ASTM standards, visit the ASTM website, www.astm.org, or contact ASTM Customer Service at service@astm.org. For *Annual Book of ASTM Standards* volume information, refer to the ~~standard's~~standard's Document Summary page on the ASTM website.

[3] Available from ~~International Electrotechnical Commission (IEC), 3, rue de Varembé, 1st Floor, P.O. Box 131, CH-1211, Geneva 20, Switzerland, http://www.iec.ch.~~Federal Aviation Administration (FAA), 800 Independence Ave., SW, Washington, DC 20591, http://www.faa.gov.

[4] Available from Radio Technical Commission for Aeronautics (RTCA), 1150 18th NW, Suite 910, Washington, DC 20036, ~~www.rtca.org.~~https://www.rtca.org.

[5] Available from SAE International (SAE), 400 Commonwealth Dr., Warrendale, PA 15096, https://www.sae.org.

~~3.1.2 *continuous built-in test*—component level tests that are critical for monitoring the integrity of data and health of the aircraft systems which are crucial for validating the data used for determining acceptable aircraft safety and stability and control.~~

~~3.1.3 *decision delay*—cumulative delays from the safety monitor and the RTA Switch.~~

~~3.1.4 *input delay*—cumulative delay from the sensed inputs and the RTA Input Manager.~~

3.4.1 ~~*non-pedigreed*~~*Assured,* ~~*components*—~~*adj*—~~hardware and software items for which the UAS manufacturer does not or cannot produce sufficient evidence that these items on their own will operate within~~ Attribute of an entity for which sufficient evidence exists to demonstrate that an acceptable level of ~~risk based on the operational risk assessment.~~rigor has been met.

3.4.2 *Complex Function, n*—the unassured function for which run-time-assurance is being used. For examples, see *Background*.

3.4.3 *Designer, n*—the person or organization that is responsible for the design, development, and/or integration of the RTA System.

3.4.4 *Dynamic Consistency, n*—independently measured variables are checked for consistency using known models of behavior. For further detail, reference ASTM AC377 TR2-EB, Section 5, Dynamic Consistency.

3.4.5 *Input Manager, n*—an assured RTA function that accepts assured and unassured data and conditions, validates and performs consistency checking, and outputs assured data to RTA Components.

3.4.6 *Larger System, n*—the system within which the RTA System exists. It provides external RTA data and inputs and consumes the RTA Output. *Example of Larger Systems are avionics system/subsystem, air vehicle, or UAS, which may contain multiple RTA Systems.*

3.4.7 *Larger System Specification, n*—The collection of all requirements used to specify the design of the Larger System. A subset of the Larger System Specification contains requirements derived from this architecture standard specifying the design and implementation of the RTA System.

3.4.8 *Monitor Coverage, n*—union of the coverage provided by each Monitor Subfunction within the RTA system.

3.4.9 ~~*pedigreed components*—~~*Predefined Bounds, n*—~~hardware and software items for which the UAS manufacturer produces sufficient evidence that these items on their own will operate within an acceptable level of~~acceptable limits to maintain the Larger System in a Safe State. Any violation of this bound is a failure of the RTA System. ~~*risk based on the operational risk assessment.*~~*Predefined Bounds may be static or dynamic and are determined during design.*

3.4.10 ~~*pre-defined limits*—~~*Recovery Function, n*—~~defined not-to-exceed restrictions that, if exceeded, would create a safety hazard.~~an assured RTA function that generates outputs intended to keep the Larger System in a Safe State. ~~*These "hard limits" are determined from the operational risk assessment (for example, taking into account vehicle characteristics, CONOPS, etc.).*~~*Recovery Function may provide "fail safe" or "fail functional" capabilities in order to allow for graceful degradation of functionality.*

3.4.11 *Recovery Function Coverage, n*—union of the coverage provided by each Recovery Function within the RTA System.

3.4.12 *RTA Components, n*—the set of assured functions defined by RTA architecture; includes Input Manager, Safety Monitor, RTA Switch, Recovery Function(s).

3.4.13 *RTA Output, n*—the output of the RTA Switch.

3.4.14 *RTA Switch, n*—an assured RTA function that accepts the source selection for the RTA Output from the Safety Monitor and provides that one output to the Larger System.

3.4.15 *RTA System, n*—the system containing RTA Components and the Complex Function.

3.4.16 *RTA System Coverage, n*—the RTA System's operational domain where both Monitor Coverage and Recovery Coverage exists.

3.4.17 ~~*recovery control*~~*Run-Time Assurance,* ~~*function*~~—*n*—a ~~pedigreed function or software algorithm to return the UAS to a safe state. For example, a sequence of commands that causes the UAS to land safely, to maneuver in space, return to level flight, or deploy a flight recovery system.~~method that uses RTA systems to ensure that a Larger System's behavior remains in a Safe State.

~~3.1.8.1 *RCF complete*—the system state where the RCF has been effective in ensuring the UAS will not violate its pre-defined limits.~~

~~3.1.8.2 *RCF delay*—the cumulative delay from each RCF.~~

~~3.1.8.3 *RCF response delay*—the delay between the initiation of the RCF and RCF complete.~~

~~3.1.8.4 *RCF trigger thresholds*—the thresholds in the safety monitor which the UAS manufacturer sets to ensure that action is taken before the UAS violates a pre-defined limit. These "soft limits" trigger the safety monitor to command the RTA switch to an appropriate RCF and account for all delays between command of the RTA switch and the execution of the recovery action.~~

3.4.18 ~~*run-time assurance architecture*—~~*Run-Time Assurance Architecture, n*—a system of ~~pedigreed~~assured components that implements ~~real-time~~ monitoring, prediction, and fail-safe recovery mechanisms that bounds the ~~flight~~ behavior of a ~~non-pedigreed complex function to ensure the safety of a UAS. Includes the components in~~system containing a Complex Function. RTA Components are: ~~Fig. 1.~~Input Manager, Safety Monitor, RTA Switch, and Recovery Function(s).

~~3.1.9.1 *RTA input manager*—a function or device that integrates sensor data and monitors sensor state.~~

~~3.1.9.2 *RTA recovery time*—the delay between the inputs to the RTA architecture and RCF complete. RTA recovery time includes the RTA response time plus vehicle dynamics, human response time (if implemented), etc.~~

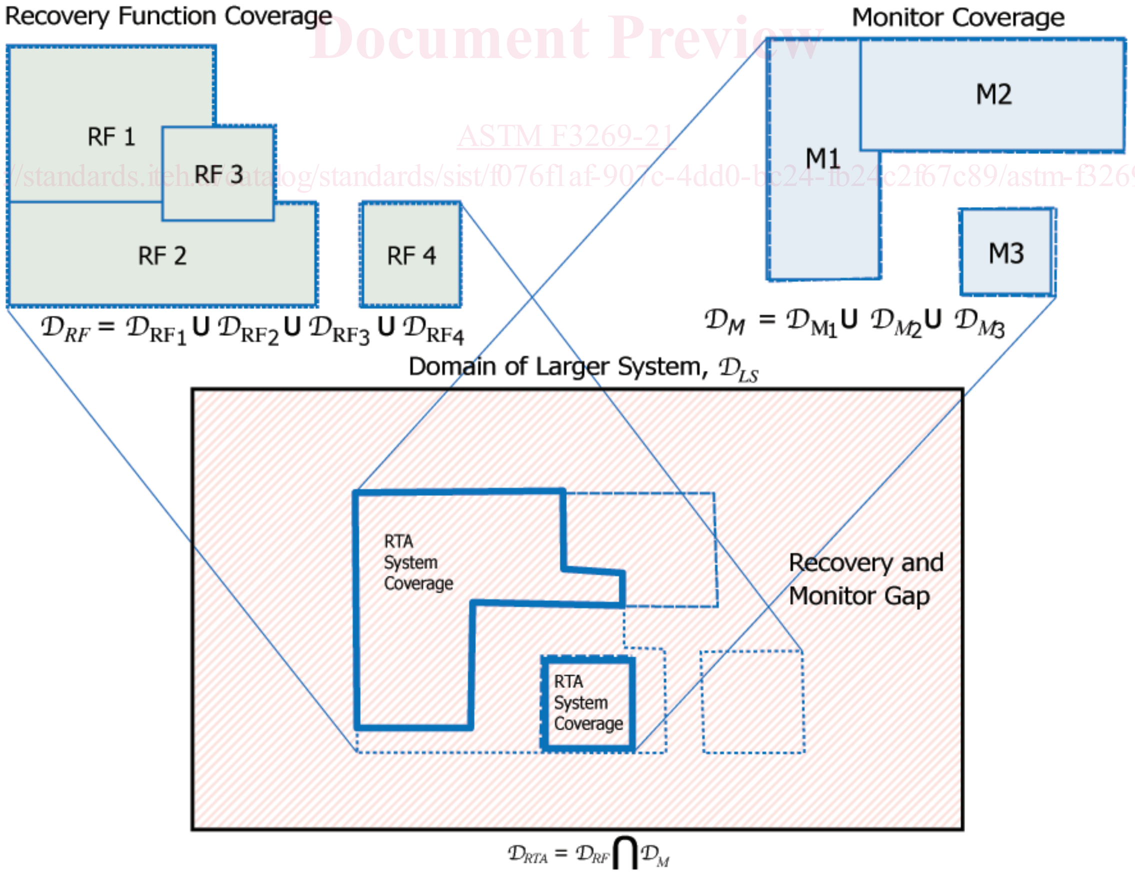
Recovery Function Coverage
Monitor Coverage
RF 1
RF 3
RF 2
RF 4
M2
M1
M3
$\mathcal{D}_{RF} = \mathcal{D}_{RF_1} \cup \mathcal{D}_{RF_2} \cup \mathcal{D}_{RF_3} \cup \mathcal{D}_{RF_4}$
$\mathcal{D}_M = \mathcal{D}_{M_1} \cup \mathcal{D}_{M_2} \cup \mathcal{D}_{M_3}$
Domain of Larger System, $\mathcal{D}_{LS}$
RTA System Coverage
Recovery and Monitor Gap
RTA System Coverage
$\mathcal{D}_{RTA} = \mathcal{D}_{RF} \cap \mathcal{D}_M$

FIG. 2 RTA ~~Response Timing Diagram~~System Coverage

~~3.1.9.3 *RTA required inputs*—data from sensors, discrete state indicators, vehicle state monitors, and other sources that describe the aircraft state and its environment.~~

~~3.1.9.4 *RTA response time*—the delay between the inputs to the RTA architecture and the activation of each recovery control function. RTA response time is the system end-to-end delay and includes input delay, decision delay, and RCF delay. See Fig. 2.~~

~~3.1.9.5 *RTA switch*—a function or device that receives control commands from the safety monitor which determines whether the complex function or specific recovery control functions are sending commands to the aircraft systems to execute the appropriate action. The RTA switch ensures that only a single function is sending commands to the vehicle management system.~~

3.4.19 ~~*safety monitor*—~~*Safe State, n—*~~continually monitors aircraft state to determine if the aircraft is or is predicted to exceed pre-defined limits. As necessary it will control the safety switch to enable execution of the recovery control function (including determining which recovery control function is executed if more than one exists).~~a condition where the Larger System is within acceptable limits.

3.4.20 ~~*shall versus should*~~*Safety Assessment Process,* ~~*versus*~~ *n—*~~*may*—use of the word "shall" implies that a procedure or statement is mandatory and must be followed to comply with this practice, "should" implies recommended, and "may" implies optional at the discretion of the supplier, manufacturer, or operator. Since "shall" statements are requirements, they include sufficient detail needed to define compliance (for example, threshold values, test methods, oversight, and references to other standards). "Should" statements also represent parameters that could be used in safety evaluations, and could lead to development of future requirements. "May" statements are provided to clarify acceptability of a specific item or practice, and offer options for satisfying requirements.~~the set of activities applied during the design of the Larger System to generate safety objectives and determine the necessary level of assurance for the RTA Components.

3.4.21 *Safety Monitor, n*—an assured RTA function that 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 a Recovery Function or back to the Complex Function. The Safety Monitor is composed of one or more Monitor Subfunctions.

3.4.22 ~~*vehicle management system*—~~*Safety Monitor Trigger Threshold (SMTT), n—*~~elements critical to maintaining normal flight and to executing all recovery control functions. Requirements for the VMS are derived from other~~limits derived from the Predefined Bounds that are used by the Safety Monitor to determine the source of the RTA Output. ~~*standards and may include certified autopilots, inner-loop flight controls, outer-loop flight controls,*~~ *The SMTT may be static or dynamic and is determined during design.*~~etc.~~

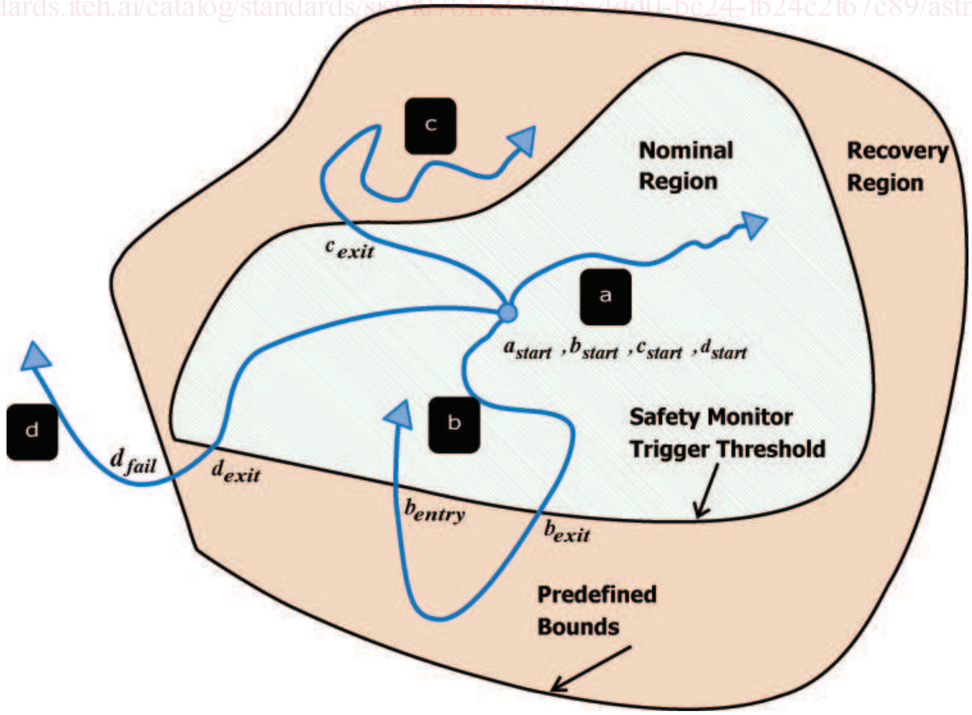

FIG. 3 RTA System Operational Scenarios

3.4.23 *Unassured, adj*—Attribute of an entity that is not assured and, hence, may not be directly used and trusted by RTA components.

3.5 ~~*Acronyms:*~~*Abbreviations:*

3.5.1 *CAA*—Civil Aviation ~~Authority.~~Authority

3.5.2 *CF*—Complex ~~Function.~~Function

3.5.3 ~~*ORA*—~~*IM*—~~Operational Risk Assessment.~~Input Manager

3.5.4 ~~*RCF*—~~*RF*—Recovery ~~Control Function.~~Function

3.5.5 *RS*—RTA Switch

3.5.6 *RTA*—Run-time ~~assurance.~~assurance

3.5.7 *SM*—Safety ~~Monitor.~~Monitor

3.5.8 *SMTT*—Safety Monitor Trigger Threshold

3.5.9 *UAS*—Unmanned Aircraft ~~System.~~System

~~3.2.8 *VMS*—Vehicle Management System.~~

~~**4. Applicability**~~

~~4.1 The focus of this practice is UAS operations, including extended visual line of sight and beyond visual line of sight operations. At the discretion of the CAA, this practice may be applied to other UAS or other aviation operations, based on a risk-based assessment of the specific aircraft design, intended mission, and area of intended operation.~~

~~4.2 The practice is expected to become an acceptable, but not the only, means of compliance in support of airworthiness, design, or operational approval processes for UAS.~~

~~4.3 The CAA requires that an applicant must show and document an acceptable means of compliance for the design and testing for the RTA system incorporated on the UAS. It is important that both the CAA and the applicant agree upon use of industry consensus standard(s) so that acceptable engineering practices are used throughout the life cycle of the product. Using these standards is intended to provide the confidence level required to allow non-pedigreed complex functions to operate while being monitored by a run-time assurance architecture for possible undetected errors.~~

~~4.4 *Run-Time Assurance Architecture*—It is assumed that the complex function will not be certified. It is assumed that the safety monitor will monitor vehicle state and command the RTA switch to a recovery control function. See Fig. 1.~~

~~4.4.1 Integrating a complex function into a UAS may result in hazards due to the complexity of the algorithm and its response to environment, mission, and vehicle state.~~

~~4.4.2 These hazards may be mitigated through the use of one or more recovery control functions.~~

~~4.4.3 The purpose of the recovery control function(s) is to ensure that hazards to third parties, other aircraft, the environment, etc. are mitigated to an acceptable level of risk.~~

~~4.5 *Run-time Assurance Architecture Description:*~~

~~4.5.1 Fig. 1 shows a minimal set of functional and input/output blocks to implement a generic RTA architecture. The RTA required inputs, RTA input manager (for example, system state, continuous built-in test, derived inputs, etc.), safety monitory, RTA switch, and recovery control function(s) all work together to monitor and limit the control authority of the complex function to maintain safety of the UAS.~~

~~4.5.2 Because of the difficulty, cost, or practically, or combinations thereof, of certifying the complex function by traditional means, the complex function is deemed to be of unknown pedigree and therefore cannot be the only available means of control to ensure flight safety. The complex function may receive input data from non-pedigreed sensor sources (these sources are not shown in Fig. 1). The RCF provides one or more alternatives to replace the control of the complex function returning the system to a safe method of control with the assurance level specified in the safety analysis. One or more recovery methods are selectable through a switch that is controlled by the safety monitor and passed to the VMS (for example, physical plant, inner-loop flight control, outer-loop flight control, etc.).~~

~~4.5.3 Recovery control functions can be either temporary (that is, control is passed back to the complex function) or terminal (that is, control remains with the recovery control function until flight is terminated).~~

~~4.5.4 The safety monitor's sole purpose is to monitor the UAS safety state (that is, the state of the UAS relative to potential hazards), to determine (via the RTA switch) which function (either the complex function or one of the recovery control functions) has control authority.~~

~~4.5.5 An example implementation of an RTA architecture for a ground collision avoidance system is provided in Appendix X1.~~

~~4.6 Fig. 2 contains a timing diagram which is a graphical representation of relationship of different measures of RTA processing (green) to various milestone events both external (dark orange) and internal (in black) to the RTA. The timing diagram is intended to be used to explain these relationships to aid UAS manufacturers in calculating the appropriate RCF trigger threshold for each RCF. The RCF trigger threshold is designed to allow sufficient time (that is, the RTA recovery time) to ensure that the RCF is able to complete its execution. The RTA recovery time includes the total processing time of the RTA (that is the RTA response time) and the execution time (that is, the RCF response) of each RCF. A number of processing delays are included in the RTA response time including the time it takes to process sensor data (that is, the input delay) the time it takes for the safety monitor to process the input data and trigger a command (that is, the decision delay) and the time it takes for the RCF to begin to maneuver the aircraft (that is, RCF delay). Each RCF delay and the RCF response has to be calculated for each RCF and is then used to define the appropriate RCF trigger threshold. Based on the operational risk assessment (for example Practice F3178), a safety buffer may be added. Section 5 contains a detail discussion of the RTA requirements.~~

4. Significance and Use

4.1 This practice provides an architectural framework for developing an RTA system, which provides run-time assurance as an alternative to design-time assurance to fulfill safety requirements for an unassured or complex function. The standard provides best practices and guidelines to assist in the RTA system's development. Further, it describes the architectural components and requirements for designing the RTA system. Compliance to this practice is achieved by deriving RTA System requirements from the standard and capturing them in the Larger System Specification. The system design requirements can then be validated and verified using acceptable engineering practices. It is anticipated that this practice will provide a means to accept complex automation/autonomy aircraft functions that have been difficult to certify using traditional methods.

4.2 The following three-step process is used to derive verifiable design requirements using this architecture standard:

4.2.1 Create RTA System requirements using the guidance provided by this architecture standard.

4.2.2 Capture RTA System requirements in the Larger System Specification.

4.2.3 Perform verification and validation on the RTA System requirements in the Larger System Specification.

4.3 The RTA architecture can be applied to all sizes, levels, and classes of UAS. Using run-time assurance can provide systems with the following benefits:

4.3.1 The ability to mitigate hazards related to nondeterministic or unexpected behavior from unassured functions that employ advanced software methods or algorithmic complexity that cannot be certified using traditional certification practices.

4.3.2 The ability to use functions for which it may not be possible to obtain artifacts of conventional DO-178 or DO-254 assurance processes.

4.3.3 The ability to use COTS hardware or software, or both, for the unassured function.

4.3.3.1 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.

4.3.3.2 For example, industry components where source code or other associated engineering artifacts are unavailable.

4.3.4 A reduction in cost and schedule burdens by allowing rapid design iterations of the unassured or complex function during and after initial certification. This update of the standard allows unassured or complex function upgrades after initial certification to minimize subsequent modifications to the certification or approval.

5. ~~Requirements~~RTA Functional Architecture

5.1 This section defines key attributes of the overall RTA architecture and its components that meet the intent described in this practice. Minimum requirements are defined for various intended uses of this reference architecture (see Fig. 1). It is expected that the reference architecture will be tailored by the users to their specific application.

5.2 Subsections 5.3 through 5.9 are written to solely provide the functional characteristics of an RTA System. The RTA components with their attributes and their relationships are described in 5.3 through 5.9. Implementations may distribute RTA functionality across hardware and software modules, as desired.

5.3 This practice is written with a single RTA System in mind, that is, where a Larger System's behavior is bound using a single RTA implementation. However, complex systems containing multiple independent RTA systems are envisioned. This practice is applicable to multiple, composable RTA systems as long as their respective RTA Outputs are independent (that is, RTA Systems

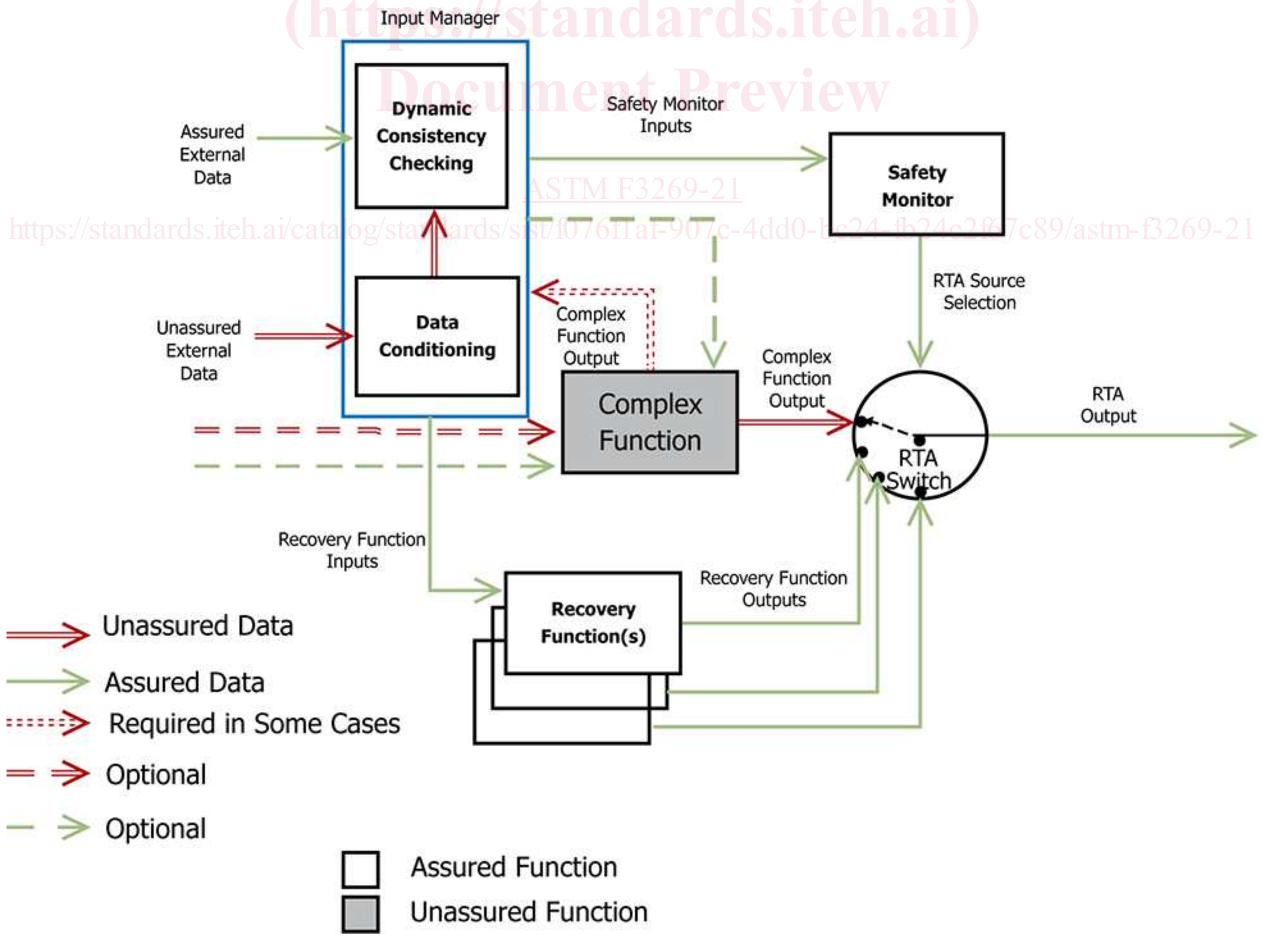

FIG. 1 ~~Functional Components of a Generic Run-Time Assurance~~ RTA Architecture do not contend to output the same data).