NASA/CR–2020-220586

Run Time Assurance as an Alternate Concept to Contemporary Development Assurance Processes

Eric M. Peterson
Electron International II Inc., Phoenix, Arizona

Michael DeVore and Jared Cooper
Barron Associates, Inc., Charlottesville, Virginia

Greg Carr
Architecture Technology Corporation, Campbell, California

April 2020

NASA/CR–2020-220586

Run Time Assurance as an Alternate Concept to Contemporary Development Assurance Processes

Eric M. Peterson
Electron International II Inc., Phoenix, Arizona

Michael DeVore and Jared Cooper
Barron Associates, Inc., Charlottesville, Virginia

Greg Carr
Architecture Technology Corporation, Campbell, California

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NNL16AA12B/0LARC18F0193

April 2020

Acknowledgments

This research work was awarded by NASA under Basic and Applied Aerospace Research and Technology (BAART) Contract No. NNL16AA12B, Task Order No. 80LARC18F0193. The research was jointly funded by NASA and the FAA. The NASA technical monitor for this task is Mr. Wilfredo Torres-Pomales. The FAA technical monitors are Barbara Lingberg and Srini Mandalapu.

The authors would like to acknowledge Jacek Kawecki and the Uber Elevate team for providing a distributed electric propulsion vehicle concept with distributed control architecture for an Urban Air Mobility concept of operations which was analyzed in the context of evaluating alternate assurance practices.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Table of Contents

Table of Figures

Table of Tables

Table of Acronyms, Initialisms and Abbreviations

Assmnt	assessment
A/C	aircraft
AC	Advisory Circular
ACAH	Attitude Command/Attitude Hold
AEH	Airborne Electronic Hardware
AFHA	Aircraft Functional Hazard Assessment (ARP4761)
AHRS	Attitude Heading Reference System
Amndmt	Amendment
AMOC	Acceptable Means of Compliance
AOA	Angle of attack
ARP	Aerospace Recommended Practice
ASMP	Aircraft Assumptions
ASTM	ASTM, 100 Barr Harbor Drive, West Conshohocken, PA 19428-2959 USA
ATC	Aircraft Type Code (ASTM)
CFR	Code of Federal Regulations
CM	configuration management
CMD	Command (Lane)
CR	Flight Crew
Cntrl	control
Cpling	coupling
CV	Control variables
DAL	Development Assurance Level
DEP	distributed electric propulsion
DI	Dynamic Inversion
DTD	Dated
EASA	European Aircraft Safety Agency
eCRM	Elevate Common Reference Model
EMA	Electromechanical actuator
Exec	executable
FCC	Flight Control Computer
FDAL	Functional Development Assurance Level (ARP4754A)
FFPA	functional failure path analysis
FHA	Functional Hazard Assessment
FLT	Flight
FPCS	Flight Propulsion Control System

FT	Feet
FTA	Fault Tree Analysis
FWD	Forward
GA	General Aviation
GW	gross weight
H	hour
HC	High confidence
HL	high level
H-Stab	Horizontal Stabilizer
HW	hardware
IDAL	Item Development Assurance Level (ARP4754A)
IFR	Instrument Flight Rules
Kg	Kilogram
Km	Kilometer
lbs	pounds
LC	Low confidence
LOC	Level of confidence
LOC	Loss of control
LL	low level
LRU	Line replaceable unit
LT	Left
MAC	Minimum Acceptable Control
MAINT	Maintenance
MCDC	modified condition decision coverage (DO-178)
MFMS	Multi-function/Multi-system (ARP4761)
MOC	Means of Compliance
MON	Monitor (Lane)
MPH	Miles per hour
MTBF	Mean time between failure
MTOW	Maximum Take-Off Weight
NEU	North East Up
NLDI	Nonlinear dynamic inversions
Obj	Object or Objective
OC	Occupants
PA	process assurance
PAX	Passenger
PFH	Per flight hour

PFP	Per flight phase
PID	Proportional-integral-derivative
PSSA	Preliminary System Safety Assessment (ARP4761)
QA	quality assurance
RCHH	Rate command / height hold
Reqt	requirement
RT	Right
RTA	Run Time Assurance
SC	Special Condition
SEC	Second
SFHA	System Functional Hazard Assessment (ARP4761)
Spec	Specification
SRS	System Requirement Specification
SSA	System Safety Assessment (ARP4761)
SSM	Signal select and monitoring
SSMP	System assumptions
SW	software
Sys	system
TOGW	Takeoff gross weight
TRC	Translation Rate Control
UAM	Urban Air Mobility
UAS	Unmanned Air System
V&V	Validation and Verification
VFR	Visual Flight Rules
VRS	Vortex ring state
VTOL	Vertical Takeoff and Landing
w/	with
WT	Wingtip

Executive Summary

NASA, in the "Effectiveness of Alternate Concepts to Contemporary Development Assurance Processes" Task sought industry research to identify and evaluate novel or alternate concepts for assuring safety of airborne systems. Using a baseline of existing industry-standard approaches to system safety assurance, this research explores alternate concepts and evaluates their effectiveness against current industry practices.

The two high-level objectives for this task order were to:

- Identify, demonstrate and evaluate the application of alternate concepts to assess and accept new technologies into existing airborne system architectures and,
- Identify, demonstrate and evaluate the ability of alternate concepts to assess and establish the airworthiness of novel, airborne system architectures.

In the context of accomplishing the two objectives, NASA desired the research to focus on a general aviation and rotorcraft type aircraft. These two vehicle categories have substantially different regulatory rule sets and compliance criteria which must be considered. The research herein therefore considers both 14CFR Part 23 Airworthiness Standards: Normal Category Airplane [3] as well as 14CFR Part 27 Airworthiness Standards: Normal Category Rotorcraft [4] as the certification basis. The project, as noted by the NASA Statement of Work (SOW), is limited in scope to primarily those regulations within the two 14CFR parts which deal with vehicle systems and equipment safety.

As summarized in Figure 1, advisory material identifies industry standards as a way to show acceptable means of compliance to the "safety regulations". The guidance consists of accomplishing objectives related to two distinct areas; development and implementation. This project captures baseline objectives for both areas to form the basis of comparison for later alternate concept research.

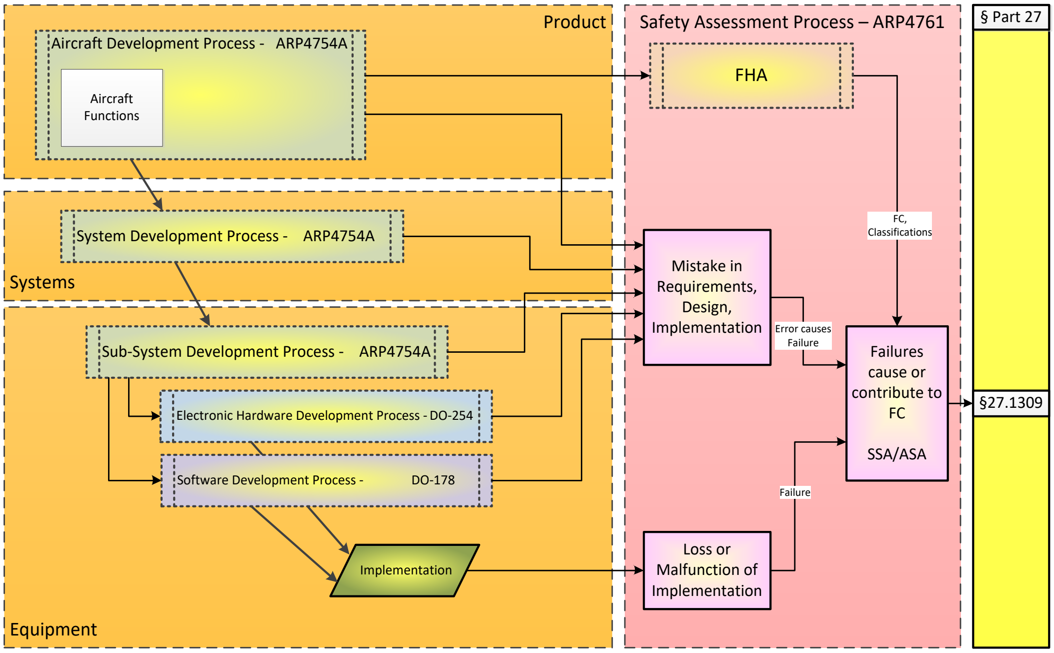

Product
Aircraft Development Process - ARP4754A
Aircraft Functions
Systems
System Development Process - ARP4754A
Equipment
Sub-System Development Process - ARP4754A
Electronic Hardware Development Process - DO-254
Software Development Process - DO-178
Implementation
Safety Assessment Process – ARP4761
FHA
FC, Classifications
Mistake in Requirements, Design, Implementation
Error causes Failure
Failures cause or contribute to FC
SSA/ASA
Failure
Loss or Malfunction of Implementation
§ Part 27
§27.1309

Figure 1. Existing Advisory Material for AMOC

In September 2018, NASA awarded Architecture Technology Corporation (ATCorp), Electron International, and Barron Associates a contract under the Basic and Applied Aerospace Research and Technology (BAART) to conduct research into the Effectiveness of Alternate Concepts to Contemporary Development Assurance Processes.

During the execution of this research effort the ATCorp Team focused on evaluation of Run Time Assurance (RTA) as an alternate concept applied to a novel, airborne system architecture. In particular the ATCorp team illustrated the application of an RTA pattern (with a high automation control mode and a low automation recovery mode) to a notional integrated flight and propulsion control system for a DEP VTOL aircraft. The general approach of RTA could also be applied to existing airborne systems and architectures.

1. Research Focus Summary

The ATCorp Team research effort included three main focal areas that corresponded with the Objectives, Tasks, and Deliverables identified in the Task Order Statement of Work (SOW):

- Identification and description of the current applicable assurance practices, along with identification and exploration of alternate assurance concepts.
- Definition of a notional airborne system to illustrate the application of current practices as well as the application of the alternate assurance concept – Run Time Assurance (RTA).
- Case Study application of baseline assurance practices and RTA to the notional system in order to illustrate the required engineering design considerations, and possible advantages and disadvantages of each approach.

A brief summary of each of the three focus areas is provided in the following subsections.

1.1. Identification of Current Assurance Practices and Alternate Approaches

Under current practices, the certification goal of assurance is to efficiently provide safety aspect coverage of systems and equipment providing complex and interrelated functions through:

1) Development assurance using a combination of process assurance and verification coverage criteria, and

2) Structured analysis or assessment techniques applied at the aircraft level to integrated and interacting systems.

The combination of these two aspects provides increased confidence in identification and correction of errors/mistakes in requirements, design, integration or interaction effects and that the implementation satisfies both the qualitative and quantitative certification criteria. To accomplish the certification goals, each aircraft standard category contains a “safety rule” and provides guidance material for safety and assurance objectives which are accomplished to show compliance to the “safety rule”. Historically, this was the “xx-1309” regulation in each regulatory part and this regulations advisory material provided the safety and assurance criteria for that vehicle systems and equipment. The associated advisory material identified the acceptable means of compliance.

From the perspective of evaluating alternate assurance concepts, it is important to establish the appropriate rule-set in order to understand the safety objectives associated with the final implementation. Each certification regulatory set contains safety and assurance objectives for implementations based on regulatory advisory material applicable to that part. How the Applicant is to address an evaluation of the final implementation as well as the development process tailored to address the severity of error or failure consequences have been described for Applicant response. As part of this study, we extracted this certification criterion in order to capture a baseline set of objectives for the current practices.

Novel aircraft designs to support Urban Air Mobility (UAM) operations may require much more complex function designs than traditional aircraft in order to accommodate the degrees of freedom provided by tilting rotors, redundant propeller/motor combinations, additional control surfaces, and other design enhancements. Moreover, operational concepts for UAM often assume that non-experts are piloting the aircraft, implying that the likelihood of manual recovery from failure conditions is remote. Thus, these systems will require a greater degree of fault tolerance than has typically been the case for Part 23 vehicles. Providing the required fault tolerance using classical development techniques will be difficult (and perhaps prohibitively expensive) due to the problem complexity. Advanced control techniques, such as adaptive control, can provide a robust and cost-effective alternative. However, adaptation relies on current (unpredictable) environmental conditions, and many learning approaches are nondeterministic in nature.

This revolution in Part 23 aircraft is coincident with a wider industry interest in more advanced aircraft systems that could exhibit greater levels of autonomy, dynamically adapt for improved performance, and/or intelligently respond to hardware faults or physical damage. Existing certification approaches are not well suited to many of these advanced designs. This is because the exhaustive analyses called for in current approaches cannot reasonably cover the entire operating space due to the algorithms' non-deterministic nature. By non-deterministic, we do not necessarily imply that the algorithms employ randomized techniques, but that they adapt in response to an environment that cannot be completely known during the design process. Such algorithms are impossible to fully analyze at design time, and it is impossible to explore, study, or simulate every possible state or outcome such systems will exhibit when exposed to the infinite possibilities of real-world scenarios, unforeseen events, and unanticipated environmental conditions.

RTA is an approach that offers a potential path for certifying these kinds of systems functions. The RTA process involves augmenting a difficult-to-certify system function with a more traditional, readily-certified system function, and in so doing significantly reducing the development rigor of the difficult-to-certify system function. The key advantage of this approach is that it provides a path for certifying non-traditional control algorithms while staying largely within the framework of established means of compliance, such as the family of specifications stemming from ASTM F3264-18b, "Normal Category Aeroplanes Certification" [15].

The outcome of the first research focus area is summarized in Appendix C.

1.2. Definition of a Notional Airborne System

The ATCorp Team developed a notional airborne system for a new and novel DEP VTOL based on Uber Elevate's eVTOL Common Reference Model (eCRM). The aircraft is a powered-lift vehicle to be developed and certificated under Normal, Utility, Acrobatic and Commuter Category Airplane regulations for VFR day use. The ATCorp Team worked with Uber Elevate to define aircraft level functions and requirements at a level of detail sufficient to support the evaluation of baseline practices and RTA during the Case Study.

It should be noted that the notional airborne system and the Case Study are focused on the processes and activities associated with identifying and applying development assurance techniques to a complex airplane function. Conventional assurance techniques as well as an alternate assurance technique are applied and discussed to facilitate other task order goals. Not all of the technical issues associated with developing the selected complex function are addressed. Design solutions developed as part of this example are notional and not representative of any planned or actual certification project solution.

The notional airborne system is captured in the Case Study Example included in Section A2.0 of Appendix A.

1.3. Case Study Application of Alternate and Contemporary Assurance Practices

This case study examines the application of current development assurance practices and an alternate assurance concept to the development of an airplane flight and propulsion control function. The airplane level function "Provide Control of Movement" and a sub-function "Provide Controlled VTOL Flight" form the framework for the discussions and examples. The case study discusses the assurance factors associated with development of the selected function in a central computer complex.

The case study certification approach has been established as 14CFR Part §21.17 [1] using the applicable regulations from 14CFR Part 23 Amendment 23-64, Airworthiness Standards for Normal, Utility, Acrobatic and Commuter Category Airplanes [3] and 14CFR Part 27 Amendment 27-35 Airworthiness Standards: Normal Category Rotorcraft [4].

Industry Consensus Standards, as advised in AC 23.2010-1, "FAA Accepted Means of Compliance Process for 14 CFR Part 23" [7], were used to accomplish the safety intent of the regulations. Certification regulation and compliance method cross reference tables capture the Part 23 regulation text and planned compliance method(s). A 14CFR Part 23 Means of Compliance (MoC) Matrix example is used to show compliance to the regulations, §23.2500, §23.2505 and §23.2510, which focus on assurance methods as the primary means of compliance.

This study example focuses on a general run-time assurance pattern, in which a highly automated control mode is coupled with a lower automation mode. The RTA goal is to reduce the time and expense associated with validating the design and verifying the implementation (V&V) of the high-automation algorithm by adding an RTA monitor and switch that prevents the high-automation algorithm from violating key safety requirements of the system. The intent is that V&V of the RTA components is easier than V&V of the high-automation algorithm.

This formulation is motivated by an assumption that UAM providers will initially deploy aircraft operated by highly trained pilots who have significant vertical-lift experience. Over time, however, these companies will transition to operations by minimally trained pilots who have no significant vertical lift experience. This will require a corresponding increase in the level of flight control automation, permitting these lesser trained pilots to handle the complexity of hover and transition flight modes associated with UAM operations. This increased automation will require certification of flight control software that is significantly different than previous Part 23 designs, and it is expected that the engineering expertise to directly produce and evaluate high-DAL implementations of this software will be lacking.

The goal of this RTA example application is to permit the gradual roll out and subsequent maturation of these higher-level control algorithms during the initial UAM business phase, while the company is employing highly trained pilots for its operations. The concept involves placing both low-automation and high-automation software on the aircraft, together with RTA monitoring and switching components. The low-automation algorithms will constitute the high-confidence system; from a certification perspective, this is the primary system. Its design, implementation, and approval processes will follow standard practices for a high DAL system, which will be based on an assumption that a highly trained pilot is operating the aircraft. The high-automation software will constitute the low-confidence system; it will be assigned a low DAL.

During flight, while the high-automation algorithms are operating, the RTA system will monitor the aircraft state for any impending violation of safety requirements. When necessary, it will switch to the low-automation software to prevent such violations. Because the pilots are highly trained, this switch in control could be reasonable, provided that the pilots are familiar with this switching behavior and such a switch is properly announced. The RTA system will be assigned a high DAL, commensurate with that of the low-automation control algorithms.

The Case Study is documented in its entirety in Appendix A.

2. Research Summary and Recommendations

The research indicates that implementing an RTA architecture approach does provide advantages in the certification of complex airborne systems. This architecting structure allows the system designers to isolate desired vehicle system functionality into low confidence and high confidence assurance strata resulting in equivalent safety results. The key element in the RTA architectures is establishing the high confidence safety boundary monitoring mechanisms which will allow the vehicle to achieve the necessary certification criteria.

2.1. Alternate Assurance Concept Equivalence Evaluation

As noted in Appendix B2.4, what industry currently identifies as baseline “objectives” are essentially activities which have been specified to result in a qualitative, undefined objective. Since the alternate assurance concept may not contain the same activity “objectives”, a different approach for comparison must be developed.

The following strategy was used to establish equivalency of the alternate concept to the current industry baseline process:

A. A qualitative level of confidence objective criteria was derived from the current industry baseline practices. The derived LOC criteria capture the system and equipment characteristics needed at assurance level for a function to operate without unintended or unexplainable operation at that level. (See Appendix B).

B. The Alternate Assurance Concept was applied to the Notional System function presented in Appendix A.

C. The activities associated with the LOC of the example implementation were captured based on the assurance level assigned.

D. The LOC achieved using current industry baseline practices activities were then compared to the LOC achieved using the alternate assurance concept.

Appendix A captures the development of a specific function within a DEP vehicle system. The specific function example was developed using a baseline approach which included current industry recommended practices and guidance. The specific function was also developed using the selected Run Time Assurance (RTA) alternate assurance concept. Development assurance and safety activities were defined and summarized for both process approaches.

Table 1 summarizes the quantity of activities to be accomplished for either baseline or RTA approach to meet the current 14CFR Part 23 certification criteria identified in Appendix B2.4. Note that RTA allows the same level of confidence to be achieved as the baseline while requiring fewer process activities to be accomplished.

Table 1 Assurance Activity Comparison Summary

Assurance Approach	Aircraft/System Level Activities	DO-178C/DO-331 SW Activities
Baseline	110	880
Run Time Assurance	110	848

Note: Results of the study activity count are not absolute and may vary dependent on actual development plan factors. Data summarized from Appendix A, Table A-11 and Table A-12.

Table 2 summarizes the quantity of activities which may be required for either baseline or RTA to meet a higher level of confidence criteria as identified in 14CFR Part 27 or the EASA SC certification criteria (see Appendix B2.4).

Table 1 and Table 2 highlight that the RTA approach provides the same level of confidence in the implementation as the baseline assurance strategy while doing so with a reduction in process activities. But real benefit of applying the RTA approach is afforded when revisions or updates to the low confidence advanced controls is to be accomplished.

Table 2 Assurance Activity Comparison Summary – High LOC Solution

Assurance Approach	Aircraft/System Level Activities	DO-178C/DO-331 SW Activities
Baseline	110	1024
Run Time Assurance	110	983

Note: Results of the study activity count are not absolute and may vary dependent on actual development plan factors.

Note: Results do not contemplate the additional complexities associated with accomplishing the activities with independence.

Under a revision or sub-sequent update to the sub-function, the activity differential is even greater since the changes to the advanced control functionality is being accomplished at a low confidence level and the RTA protection monitoring functionality would not need to be re-validated or re-verified. A postulated scenario was entertained during the contemplation of the very high confidence deltas for a function update. The delta may equate up to a seven-fold decrease in assurance activities to be accomplished when RTA has been applied (see section A6.1.2).

The RTA alternate assurance approach will allow UAM companies to gradually improve their high-automation algorithms as they gain experience through real-world operations without a need to re-validate and re-verity the error mitigation monitoring mechanism for each of these new algorithm versions. This will significantly lower the development activities and reduce the time necessary to deploy updated control algorithms. It will also provide an opportunity to collect large quantities of real-world data on the behavior of the high-automation algorithms, which can be leveraged if/when they are later certified as primary systems for use by lesser trained pilots.

2.2. Recommendations

During the course of execution on this task order, additional investigation topics were identified which may further the application of run time assurance or address more details in the application of the RTA concept.

These suggested investigation topics include:

1. Explore application of RTA to other control subsystems;
   This report focuses on the use of RTA in the context of an auto-land subsystem. In an aircraft design of this nature, multiple control subsystems operate simultaneously, at different levels within a control hierarchy. These include actuator controllers, actuator allocation (mixing), one or more levels of inner-loop control (e.g., ACAH/TRC), one or more outer-loop control subsystems (e.g., guidance, collision avoidance, etc.), and mission management subsystems (e.g., routing, path planning, etc.). Additional research is warranted to explore the possible applications of RTA to these levels of control. Research is also warranted to explore issues related to the simultaneous operation of multiple RTA-protected control subsystems.

2. Explore application of RTA to other aircraft functionality;
   The example in this report focused on RTA application in aircraft flight control. However, RTA is a general assurance concept that may benefit other aircraft functionality. Additional research is warranted to explore what and how other aircraft applications may benefit from this assurance concept.
3. Continue exploration of RTA application to DEP Notional System Example;
   Continue the investigation initiated in this task order on aircraft control function development to explore how RTA application would work in practice. Investigate how the certification authority and applicant would complete the certification project. Identify the activities and processes that would complete RTA application for certification.
4. Continue development of example function detail;
   The example flight control function herein was developed to a level necessary to complete the task order objectives of assurance activity comparisons. A more comprehensive control function, captured using model based engineering techniques should be contemplated with the associated RTA boundary development to further explore the comparable assurance activity conclusions.
5. Investigate RTA Monitoring Validation;
   The RTA monitoring mechanisms (Safety Boundaries I, II & III) are key to the successful application of the RTA concept. The definition and an initial application have been worked in this task order. There is a need to further explore the development of the boundaries and establish means to validate the boundaries are the correct boundaries for the vehicle. Various techniques should be investigated including formal methods.
6. Compare development activity efforts of RTA Monitoring and RTA Monitor Validation to conventional assurance activity efforts;
   The primary benefit, as identified in this work, is that a lower confidence function is mitigated during run time for any unresolved errors. A high level activity comparison was carried out which identified this savings. During this work, it became apparent that the formulation of the RTA monitoring boundaries will require substantial effort to accomplish. Additional study should be accomplished to establish the magnitude of the boundary definition effort.
7. Implications of RTA in Very High Confidence applications;
   A cursory impact on the assurance activities required to certificate to a very high confidence level was accomplish by this task order. Additional study is recommended to further investigate the impacts associated with this certification criterion. Establishing “break-even” application criteria may be investigated to guide optimum usage of RTA in this form.

3. Alternate Assurance Concept Identification

Based on prior research for NASA and other government agencies, the ATCorp Team used Run Time Assurance (RTA), in conjunction with current assurance practices, as the alternate assurance concept to address the challenges associated with:

1) Accepting the introduction of new technologies into existing airborne system architectures, and

2) Establishing airworthiness of novel, unprecedented airborne system architectures.

The Author's exploited the team's prior experience in the development and application of RTA systems. In particular, we draw on work that was performed for the Air Force over a three-year period as part of a Phase III SBIR project that focused on overcoming technical hurdles in the implementation of RTA for highly complex safety-critical systems [43]. In this SBIR work, the authors noted that current assurance practices are focused on Design Time Assurance (DTA), while emerging technologies and capabilities will likely require the use of Run Time Assurance (RTA) systems as a means of assuring safe operations. In [43], the following two definitions were developed:

Design Time Assurance:

The process of performing V&V analysis and testing of software offline, at design time (that is, before live operation) to the level required for certification of the plant or system the software is housed on. The required certification level depends on the defined criticality of the software, which is determined by the level of hazardous effects that errors in the software can have on the plant. If a set of software can be fully V&V'd to its defined required certification level at design time, then that software is considered trusted for live operation on the plant. That is, there is an acceptable level of confidence that the software will operate correctly to the required certification level of the system within which it operates. If a set of software cannot be fully V&V'd to its defined required certification level (due to its complexity, non-determinism, etc.), then that software is considered untrusted (or does not have an acceptable level of confidence) for live operation on the plant. Untrusted software may also arise during developmental testing stages due to its experimental nature, in which full V&V analysis/testing would be cost prohibitive at that point in the design cycle.

Run Time Assurance:

The process of monitoring a system, containing untrusted software, during runtime or live operation of the plant to determine if the untrusted software is operating correctly. If it is determined that the untrusted software is not operating correctly or anomalous or unsafe behavior is detected, then to mitigate any adverse effects that may ensue, operation of the untrusted software is terminated and control is switched to trusted reversionary counterpart software. That is, the RTA system activates some type of recovery action to ensure continued safe or correct operations. The presumption here is that the reversionary software has equivalent basic functionality as the untrusted software, but would not have all of its advanced capabilities. The reversionary software would be able to continue safe operation of the plant or system, but at a reduced level of performance, capability, or functionality.

In the Final Report for the SBIR research, the concept of an RTA-protected or a runtime protected system [43] was described:

A runtime protected system is considered to be the fundamental element in an overall RTA design. This is presented here in a universal or generalized manner, applicable to any feedback level (i.e., inner-loop control, outer-loop guidance, etc.) or any system in general.

3.1. Run Time Assurance (RTA) Concept

The prior work on RTA for the Air Force focused on specific RTA approaches to single and multiple linked control systems. More generally, the RTA concept is based on a structure of a low confidence, untrusted functional element being operationally evaluated by a high confidence, trusted functional element. Shown in Figure 2, the outputs from the run time protected function operation are safe function outputs. Inputs and system function feedback provide the desired operational characteristics such that at any time during operation, the desired “safe” operating characteristics are maintained.

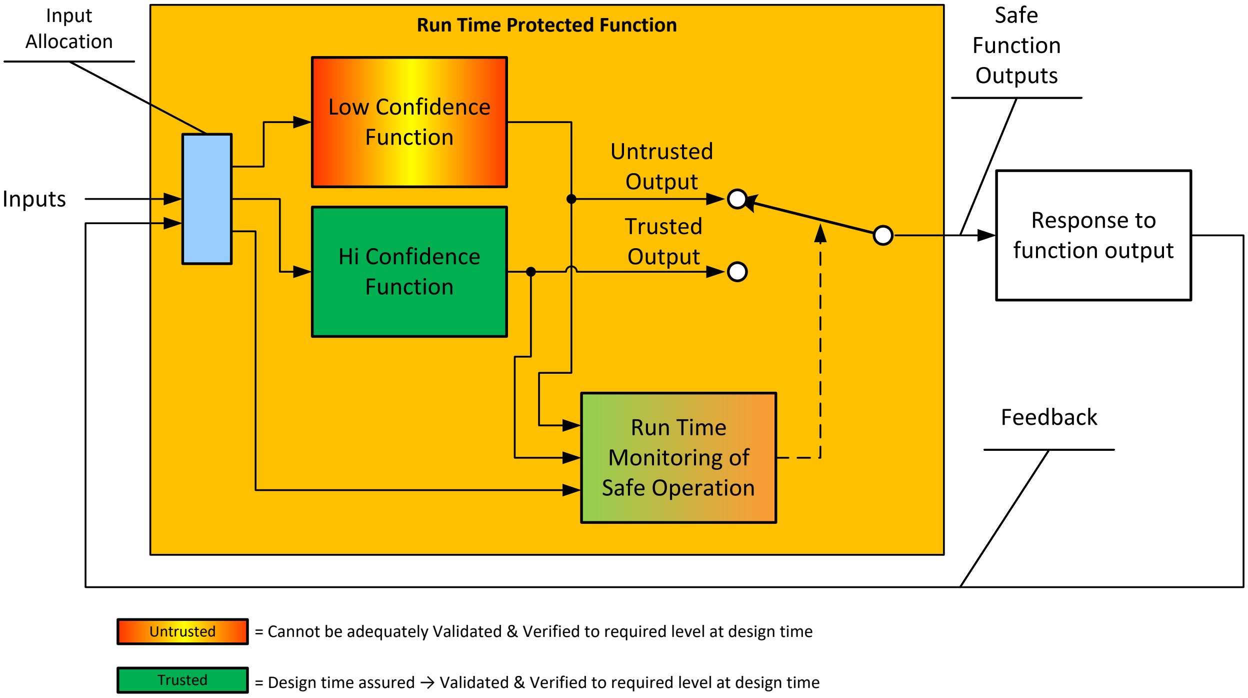

Input Allocation
Run Time Protected Function
Low Confidence Function
Hi Confidence Function
Run Time Monitoring of Safe Operation
Inputs
Untrusted Output
Trusted Output
Safe Function Outputs
Response to function output
Feedback
Untrusted
= Cannot be adequately Validated & Verified to required level at design time
Trusted
= Design time assured → Validated & Verified to required level at design time

Figure 2. Run Time Assured Operation

As noted in Figure 2, the low confidence function and its output are untrusted. This is due to the low confidence function not being fully validated and verified (V&V’d) at design-time to the same assurance level as the RTA-protected block that contains it. All of other blocks in the figure are design-time assured. That is, they are fully V&V’d to their defined requirements at the assigned assurance level during function development. Hence, all other information flow is considered trusted. All input information into the RTA protected block is trusted, and, most importantly, all information out of the RTA protected block is trusted. Even if the output of the low confidence function is passed through to the output (because no adverse conditions were detected at the current time), the claim here is that that output is trusted because it has been checked by the trusted RTA monitor.

Ultimately, the goal is to make a safety case argument that the RTA protected system is equivalent in terms of safety to a system that is fully V&V’d at design time with accepted analysis and testing practices. That is, even though there is an untrusted element within the RTA protected block, all output downstream from that block will be fully trusted information.

There are three main process elements in the application of a RTA concept. They include;

1. Establish RTA operational philosophy and goals.
2. Allocate system functionality to high confidence and low confidence development paths.
3. Define the run time fail safe boundaries and monitoring mechanisms.

The following subsections elaborate on each of the RTA process activities.

3.1.1. Establish RTA Operational Philosophy and Goals

Novel aircraft designs to support UAM operations may require much more complex function designs than traditional aircraft in order to accommodate the degrees of freedom provided by tilting rotors, redundant propeller/motor combinations, additional control surfaces, and other design enhancements. Moreover, operational concepts for UAM often assume that non-experts are piloting the aircraft, implying that the likelihood of manual recovery from failure conditions is remote. Thus, these systems will require a greater degree of fault tolerance than has typically been the case for Part 23 vehicles. Providing the required fault tolerance using classical development techniques will be difficult (and perhaps prohibitively expensive) due to the problem complexity. Advanced control techniques, such as adaptive control, can provide a robust and cost-effective alternative. However, adaptation relies on current (unpredictable) environmental conditions, and many learning approaches are nondeterministic in nature.

This revolution in Part 23 aircraft is coincident with a wider industry interest in more advanced aircraft systems that could exhibit greater levels of autonomy, dynamically adapt for improved performance, and/or intelligently respond to hardware faults or physical damage.

Existing certification approaches are not well suited to many of these advanced designs. This is because the exhaustive analyses called for in current approaches cannot reasonably cover the entire operating space due to the algorithms' non-deterministic nature. By non-deterministic, we do not necessarily imply that the algorithms employ randomized techniques, but that they adapt in response to an environment that cannot be completely known during the design process. Such algorithms are impossible to fully analyze at design time, and it is impossible to explore, study, or simulate every possible state or outcome such systems will exhibit when exposed to the infinite possibilities of real-world scenarios, unforeseen events, and unanticipated environmental conditions.

RTA is an approach that offers a potential path for certifying these kinds of systems functions. The RTA process involves augmenting a difficult-to-certify system function with a more traditional, readily-certified system function, and in so doing significantly reducing the development rigor of the difficult-to-certify system function. The key advantage of this approach is that it provides a path for certifying non-traditional control algorithms while staying largely within the framework of established means of compliance, such as the family of specifications stemming from ASTM F3264-18 [15], "Normal Category Aeroplanes Certification."

As illustrated in the control example presented in Figure 3, in the RTA approach, the non-traditional control algorithm is designated as a secondary system for the purposes of certification. (Depending on the application, the role of this algorithm may be further reduced to that of non-required equipment, with a commensurate decrease in certification effort.) The primary control system comprises a more traditional control algorithm, together with a run-time monitor and switch mechanism. This mechanism accepts outputs from both control algorithms and, based on the state of the aircraft, chooses one set of outputs to pass through to the aircraft systems being controlled. In the RTA approach, the entire primary control system, including both the traditional control algorithm and RTA monitor/switch, is assigned a DAL consistent with the failure condition classification of the primary control function. The secondary control system is assigned a lower DAL or may not have an assigned DAL.

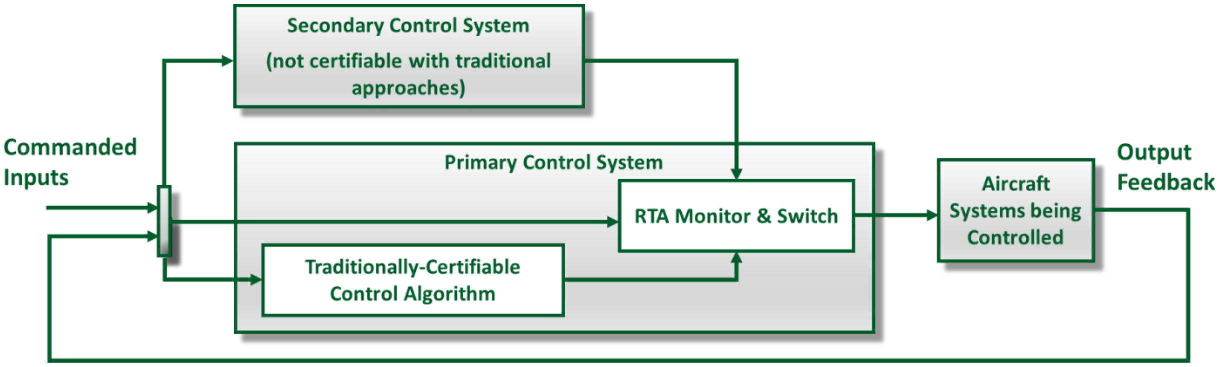

Secondary Control System
(not certifiable with traditional approaches)
Commanded Inputs
Primary Control System
RTA Monitor & Switch
Traditionally-Certifiable Control Algorithm
Aircraft Systems being Controlled
Output Feedback

Figure 3. Example Primary & Secondary Control Systems using RTA Concept

The key idea behind the RTA approach is that the RTA monitor and switch work to ensure an equivalent level of safety for the combined primary/secondary system relative to a traditional engineering and certification process. This is accomplished by continually verifying at run-time that the aircraft state is consistent with a rigorous definition of safe flight that was established at design-time, as part of the RTA process, and selecting between the two control algorithms as appropriate. In doing so, the RTA approach offers protection against both algorithm design errors and software development errors associated with the secondary controller.

An RTA process may take multiple paths, depending on the capabilities of the secondary controller, the reasons for incorporating it into an aircraft design, and the assignment of system requirements to both the primary and secondary controllers. Appendix C: Run-Time Assurance Background contains an extensive literature review of RTA applications that collectively illustrate a wide variety of function allocation strategies.

In general, there are two basic variations:

- Secondary as Main: In these designs, the secondary control algorithm is intended for use under nominal conditions, which are assumed to prevail most of the time, and the primary control algorithm is used as a backup. This is the variation that appears most common in the literature and is assumed in the standard ASTM F3269-17 [19], "Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions."[1]
- Primary as Main: In these designs, the primary control algorithm is intended for use under nominal conditions, and the secondary control algorithm is used as a backup.

It is worth emphasizing here, that the designation of primary or secondary does not refer to the operational role of a system nor does it refer to its relative frequency of use. Rather, the primary or secondary terms refer to the role of these systems in meeting the safety objectives for certification. FAA Advisory Circular AC 23.1309-1E [6] provides a relevant example for this interpretation of primary versus secondary system, albeit one that does not involve RTA.

An extract from Section 8.ii of AC 23.1309-1E document reads:

For example, a brake control system normally uses the electronic brake system most of the time because of its better performance, but it does not comply with all the requirements. In this case, the mechanical brakes are used as the backup systems; yet, it is consider (sic) the primary with regard to meeting the requirements and the electronic brake system is the secondary.

Examples of designs incorporating the secondary-as-main variation include:

- A company might want the flexibility to frequently deploy updated software to the fleet to improve performance or handling qualities over time. In this case, the original certified control algorithm that was in place when the aircraft was initially developed could be used as a backup control algorithm. Updated software, which might not have been certified for use on the aircraft, could be distributed for use as the main control algorithm.
- An aircraft design might employ two gain scheduled controllers implementing two very different schedules. The main control algorithm might be highly scheduled to deliver very efficient operations, but as a result be difficult to achieve a high enough DAL due to the complexity associated with analyzing all possible schedule transitions. It might be paired with a similar controller implementing a much lower scheduling, which could be developed to a high DAL.
- An aircraft design might incorporate an adaptive algorithm that automatically tunes for high-performance operation as the main controller. It might be paired with a traditional gain-scheduled backup controller that could be developed to the requisite DAL.

[1] Note that the terminology of "primary vs. secondary" systems and "main vs. backup" usage differs from terminology in ASTM F3269-17. That standard assumes a specific RTA use case, and the language it adopts is not suitable for all RTA cases. The "Complex Function" referred to in F3269-17 is a secondary control system used as the main controller. Similarly, the "Recovery Control Functions" in that document collectively constitute a primary control system that is used as a backup controller.

Since the backup system is trusted and certified at design time, it will typically not have greater capabilities that the main system possesses but should have the minimum required capabilities to ensure safe operation and, for example, return the platform to a safe location for recovery and post-mission analysis. In this manner, RTA protection bounds the behavior of the untrusted, main control system, allowing it to operate and provide the benefits of its advanced capabilities, but disallowing any unforeseen, unsafe actions that could compromise system safety.

Designs incorporating the primary-as-main variation typically involve damage-adaptive control or allocation algorithms as the backup controller. These would attempt to safely control the aircraft in response to significant physical damage that is beyond the capabilities of the design-time certified main controller. In this arrangement, the backup controller could be employed in a last-ditch effort to spare the aircraft and its occupants in the face of unforeseen events. This arrangement might be suitable, for example, in an aircraft for which a traditional gain-scheduled controller, used as the main controller, could ensure the failure likelihoods required for certification but some additional safety margin was desired. In this case, the adaptive backup controller may constitute Non-Required Safety Enhancing Equipment (NORSEE), as contemplated in FAA Policy PS-AIR-21.8-1602 [8].

3.1.2. System Function Allocation

When contemplating an RTA approach, several important questions must be addressed that will guide the approach and dictate fundamental capabilities of the components shown in Figure 2. These questions include, but are not limited to, the following:

- What are the functional algorithms to be implemented in the high confidence and low confidence functions? How will these functional algorithms complement each other to ensure that all aircraft design requirements are met?
- Does the benefit of including the low confidence function outweigh the resulting complication to the design, implementation, and certification activities? In other words, is there a compelling reason for including the low confidence function rather than relying solely on a single high confidence function?
- Which function (low confidence or high confidence) will be “main”, to be used under nominal conditions, and which will be the backup, to be used when an off-nominal condition has been detected?
- What are the off-nominal conditions that should result in a switch to the backup function algorithms?
- What are the means for determining that off-nominal conditions are imminent? (Note that this implies an ability for the RTA monitor to account for future states of the aircraft and its systems; it must act before safety has been compromised.)
- What system states and/or measurements carry the information necessary to make this determination? Can these states and/or measurements be delivered to the RTA monitor with high enough reliability?
- Is there an intention to take any certification credit for the low confidence function? If so, what DAL is anticipated for it, and will that be sufficient given the DAL requirements of SAE ARP4754A [22] and/or Table 2 of ASTM F3061-17 [16]?

Regardless of the specific design that emerges, the RTA process must ultimately establish that the RTA protected function performs its intended function and is absent from any unintended function. Moreover, it must establish that “there is a logical and acceptable inverse relationship between the average probability and the severity of failure conditions,” as required by 14 CFR 23.2510 [3]. The depth of analysis employed to establish these properties must be commensurate with the DAL assigned to the RTA protected function. This analysis must address the design and implementation of each component within the RTA protected function, as well as of the overall design, assuming the worst-case behavior of the low-confidence function.

It must include:

- Failure analysis, to ensure that there is no single point of failure for the RTA protected function
    - Is the output of the RTA monitor correct at all times and in all anticipated conditions? An incorrect output from the RTA monitor could lead to an incorrect output from the RTA protected function by failing to command a switch between the low- and high-confidence functions when appropriate.
    - Does the switching mechanism itself always behave properly given the output of the RTA monitoring component? A failure of the switch to transmit the correct output could result in an incorrect output from the RTA protected function.
    - If no certification credit is taken for the low confidence function, the high confidence function must be able to achieve the failure probability requirements of F3061 Sec. 4.2 [16] and the requirement in F3230 Section 4.2.4.3 [17] that no single component failure should result in a catastrophic failure condition.
- Common-mode failure analysis, to ensure that failures of the low-confidence function do not result in failures of the overall RTA protected function.
    - Are there any potential common-mode failures between the high confidence and low confidence functions? Such a failure could result in an incorrect output from the overall RTA protected function.
    - Are there any potential common-mode failures between the RTA monitor and the low confidence function? Such a failure could prevent the RTA monitor from detecting that it is necessary to activate the switching component, resulting in an incorrect output from the overall RTA protected function.
- Independence analysis, to ensure that the inclusion of an RTA protected function does not result in unwanted functions or interactions with other aircraft systems.
    - Can other critical aircraft systems be affected by faults of the low-confidence function?
    - Can the RTA monitor/switch mechanism interfere with the nonfunctional attributes such as availability, integrity, safety, performance, etc. of other flight critical systems?
    - Can the RTA monitor/switch mechanism introduce any new hazards because of its own function? For example, such hazards might arise from scheduling problems, live-lock, deadlock, lost links and other timing or communication issues.

- Can the transition between high- and low-confidence functions introduce any hazards to continued safe flight?

The RTA monitor and switching mechanism introduces several unique challenges to the analyses called out above. These analyses must specifically account for:

- the amount of time it takes the RTA Monitor/Switch to detect that it is appropriate to switch controllers and perform that switching operation;
- any transients associated with a switch between controllers;
- the accuracy, precision, availability, and timeliness of all input data upon which the RTA monitor/switch will make a switching decision;
- the dynamics of the RTA switch/monitor and all downstream and feedback systems;
- the timing and frequency requirements of all components, including downstream and feedback systems; and
- any delay arising from the RTA monitor/switch.

3.1.3. Define Fail Safe Boundaries

At the center of the RTA approach is a monitor that decides which output should be active at any given time. A formal definition of its behavior is crucial to ensure that the RTA protected function produces the correct output under all anticipated conditions. This definition must be derived from system safety and performance requirements at design time. Methodologies for producing this switching condition have been studied in most detail for the specific case of the Secondary-as-Main controller scenario.

In [43], the authors outline a process for constructing a switching condition that can be implemented in the RTA monitor and switch mechanism to ensure that the aircraft does not enter an unsafe state. That process requires designers to define a nested sequence of aircraft states that satisfy increasingly stringent notions of safety. This process is summarized in the following steps:

a) Define the set, S, to be the set of all states of the system function. This step requires designers to determine the collection of all variables related to the aircraft that play a role in its safety. That is, it includes not just the typical dynamic states that may be used to build a simulation model, but also all plant configuration states and all environmental states as well.

b) Define the set $S_{safe} \subseteq S$ to be the set of all states that are ***safe*** with respect to the system function. Safe states are operating points in which the system functions as intended. This step requires designers to explicitly determine what it means for the system to function “as intended” and how it relates to system state.

c) Construct the set $S_{Dsafe} \subseteq S_{safe}$ as the set of all states determined to be safe with respect to the system function. Note that it is often the case that the set S_{safe} cannot be precisely determined or defined. That is, the exact border separating safe from unsafe states is typically uncertain due to modeling errors, inaccuracies in measuring or estimating system states, changing environmental conditions, etc. For this reason, designers typically add in safety margins to account for such uncertainties; these safety margins are often a matter of judgment. The set S_{Dsafe} accounts for these margins and comprises the set of states that, by definition, the aircraft must never depart. Keeping the aircraft within this safety set is the primary goal of the RTA monitor and switch mechanism. As such, a point x_0 in S_{Dsafe} is said to be Type I safe. Figure 4 gives a graphical illustration of the sets S, S_{Dsafe} and S_{safe}. Note that these sets are functions only of the aircraft system function and are not dependent on either of the specific algorithms or their corresponding implementations.

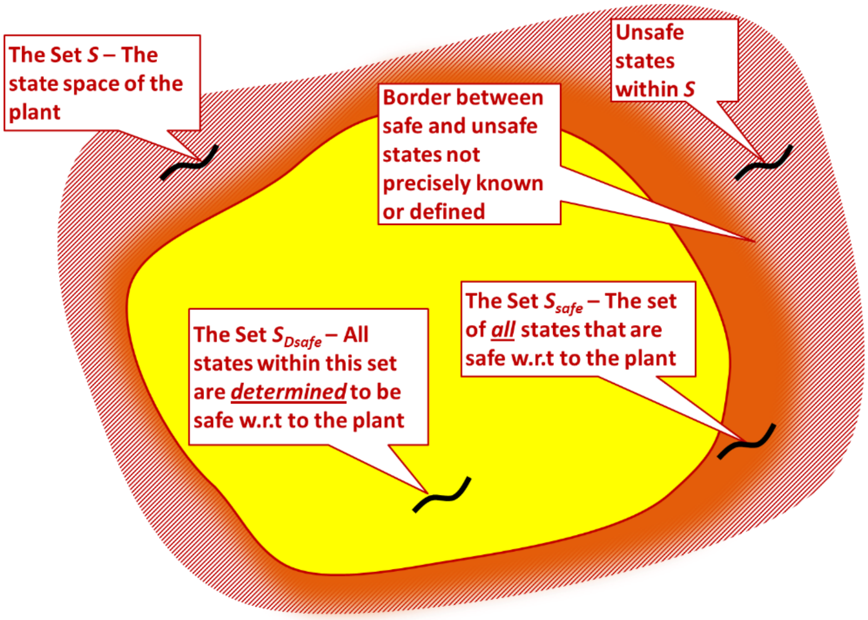

The Set S – The state space of the plant
Border between safe and unsafe states not precisely known or defined
Unsafe states within S
The Set S_{safe} – The set of *all* states that are safe w.r.t to the plant
The Set S_{Dsafe} – All states within this set are *determined* to be safe w.r.t to the plant

Figure 4. Set of States Defined to be Safe [43]

a) Determine the recovery operation that the backup control algorithm will perform if the main control system is switched out by the RTA monitor and switch mechanism. Specify this recovery operation in terms of a set $Q \in S_{Dsafe}$ and a time interval T having the property that, following a switch from the main controller to the backup controller, the backup controller will attempt to drive the system being controlled into a state inside Q within a time interval $T > 0$. The quantities Q and T may be dependent on the state of the aircraft system at the time the switching activity occurs. We define the region Q to be that set of states at which the *off-nominal condition* (which triggered the RTA monitor/switch) is be resolved or alleviated; it need not be the final state region in the recovery process. Once the state reaches Q, then it will be in a stable, safe region of attraction.

b) Determine the Type II safety region as the subset of states in S_{Dsafe} such that, upon switching to the backup controller, the state trajectory can converge to at least one point in Q within the desired time T, and that trajectory is entirely contained within S_{Dsafe}. Type II safety accounts for the behavior of the combined system consisting of the plant and the reversionary system. Its definition is such that, from any point in the Type II safe region, the RTA switch could activate the reversionary system, which can then maintain control over the plant, driving it to a desired region, *Q,* in the state space in a finite amount of time *T*. Physical systems have momentum, so state trajectories cannot always be turned instantaneously to another direction and therefore may exhibit overshoot. Further, control mode switches often result in transient behavior. So, by definition, Type II safety requires that any resulting overshoot or transients involved during the switching process and transition to *Q* will not compromise plant safety (will not leave the set S_{Dsafe}).

c) Determine τ, the time horizon used for planning purposes by the RTA monitor and switch mechanism. In most cases, these components will be implemented as discrete-time system that repeatedly checks the safety properties of the current aircraft system state, then makes a decision about whether or not to switch to the backup controller. The quantity τ must be greater than the maximum elapsed time between these successive checks.

d) Determine the Type III safety region as the set of states that are Type II safe and that have the property that every possible output of the main controller for a time period τ results in a state trajectory entirely contained within the Type II safe region. From any point in the Type III safe region, the RTA switch can pass any commands from the advanced system to the plant for at least τ seconds without exiting Type II safety.

After these determinations have been made, the RTA monitor and switch mechanism can be designed to switch from the main controller to the backup controller any time the aircraft state is found to be outside the Type III safety region. The Type I, II and III safety regions are illustrated in Figure 5.

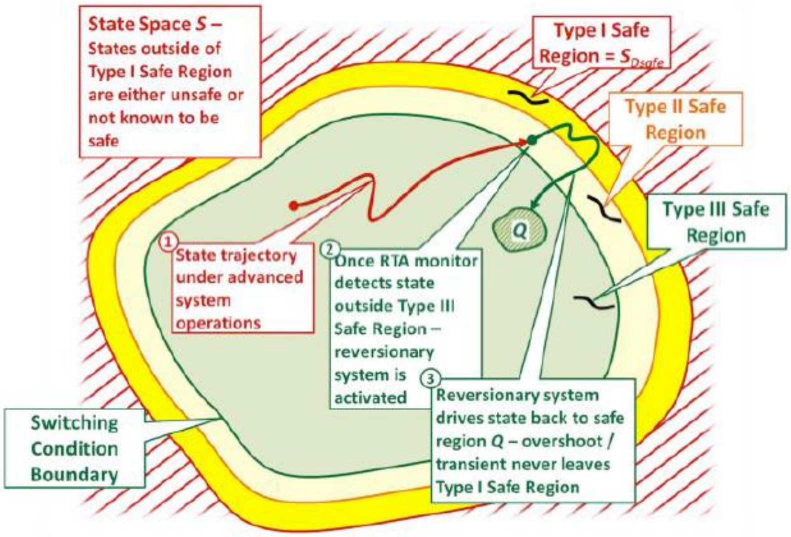

State Space *S* – States outside of Type I Safe Region are either unsafe or not known to be safe
Type I Safe Region = S_{Dsafe}
Type II Safe Region
Type III Safe Region
Q
1 State trajectory under advanced system operations
2 Once RTA monitor detects state outside Type III Safe Region – reversionary system is activated
3 Reversionary system drives state back to safe region *Q* – overshoot / transient never leaves Type I Safe Region
Switching Condition Boundary

Figure 5 Type I, II & III Safety Regions [44]

These definitions are perhaps more readily understood from the perspective of the RTA monitor's decision logic. The RTA monitor and switch component must guarantee that the aircraft never leaves the Type I safe region, which ensures that the aircraft continues to operate safely. To guarantee this, every τ seconds, it checks to see if the current state is inside the Type III safety region. If the state is in this region, the RTA monitor can allow the advanced system's commands to pass through to the plant. In the worst case, on its next check τ seconds later, the plant's state will have transitioned out of the Type III region, but it will not have transitioned beyond the Type II safe region (part of the definition of Type III safety). In that case, the RTA component must switch to the reversionary controller. Because the aircraft is now in the Type II safe region, the resulting transient will never leave the Type I safe region (part of the Type II safety definition) and the reversionary system can successfully maintain safe control/management of the aircraft.

3.2. Tailoring the Alternate Assurance Concept

The RTA approach is specifically designed to facilitate the application of traditional certification processes to non-traditional systems. As such, tailoring the level of development rigor in an RTA process is identical to that of current processes. More specifically, the RTA process predominantly affects the up-front activities of requirement derivation, function allocation, hazard classification, and DAL assignment. Once DALs have been assigned to the RTA protected function and its components, the development process proceeds as in a traditional approach, such as specified by standards like DO-178C [25], DO-254 [26], etc.

The method for tailoring the assurance rigor of the alternate concept will be developed during the Case Study applying the alternate concept to a notional airborne system (SOW Task 3.2).

3.3. Alternate Assurance Concept and Consensus Standards

The RTA alternate concept relies upon the current industry standards and practices to develop the high confidence operational function and to develop the run time monitoring and switching mechanism. The high confidence path will therefore accomplish the current industry assurance objectives and objective activities at an appropriate level of process rigor (FDAL or IDAL).

The identification of how the alternate assurance concept supports ASTM General Aviation Consensus Standards will be performed during the Case Study applying the alternate concept to a notional airborne system (SOW Task 3.2).

Appendix A: Alternate Assurance Concept Application Case Study

Disclaimer

This example is focused on the processes and activities associated with identifying and applying development assurance techniques to a complex airplane function. Conventional assurance techniques as well as an alternate assurance technique are applied and discussed to facilitate other task order goals. Not all of the technical issues associated with developing the selected complex function are addressed. Design solutions developed as part of this example are notional and not representative of any planned or actual certification project solution.

A1.0 Case Study Framework

The assurance example developed in this appendix follows the general aircraft and systems hierarchical development paradigm. Figure A-1 presents the hierarchical development activities described herein. Conventional development task activities are presented in blue boxes with safety activities presented in yellow octagon shapes. The case study example is developed “down” to the handoffs to the implementation layer.

The alternate assurance concept, Run Time Assurance (RTA), is presented in green bars paralleling the conventional development activities. The RTA concept is compatible with the conventional process and will be shown to focus on using the conventional architecture constructs, in conjunction with a unique monitoring strategy, to allow application of lower confidence implementation solutions.

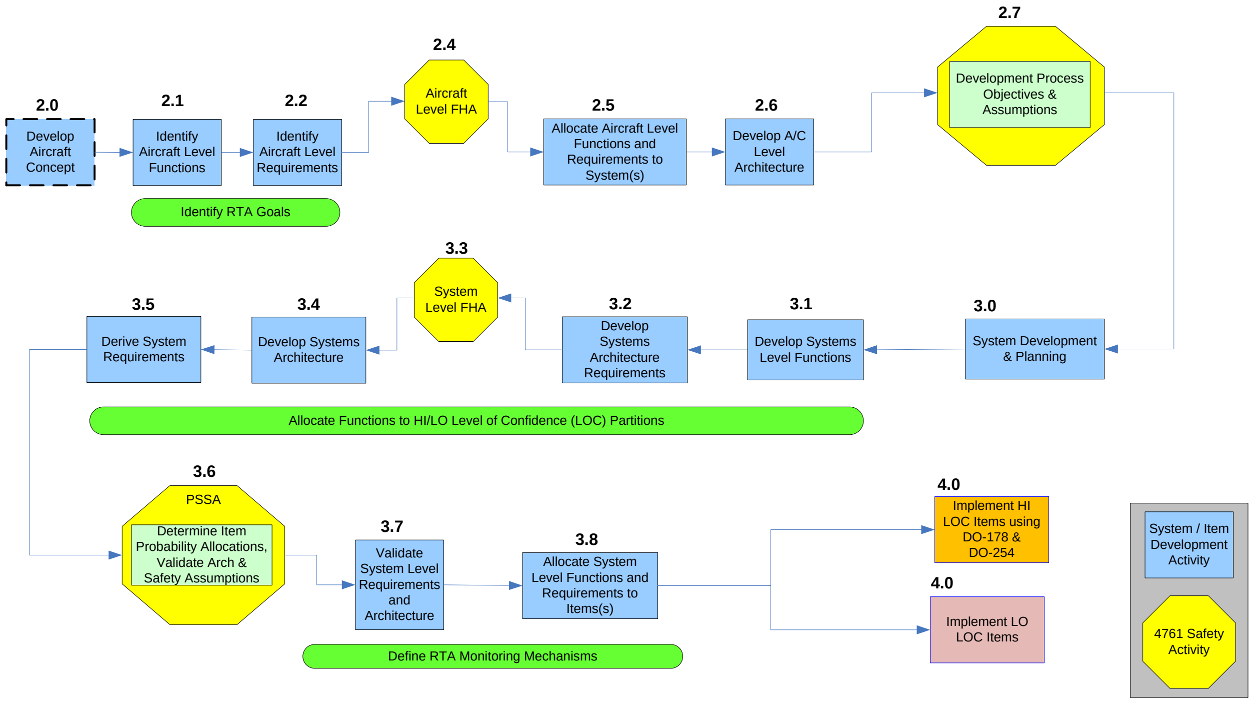

2.0
Develop Aircraft Concept
2.1
Identify Aircraft Level Functions
2.2
Identify Aircraft Level Requirements
2.4
Aircraft Level FHA
2.5
Allocate Aircraft Level Functions and Requirements to System(s)
2.6
Develop A/C Level Architecture
2.7
Development Process Objectives & Assumptions
Identify RTA Goals
3.3
System Level FHA
3.5
Derive System Requirements
3.4
Develop Systems Architecture
3.2
Develop Systems Architecture Requirements
3.1
Develop Systems Level Functions
3.0
System Development & Planning
Allocate Functions to HI/LO Level of Confidence (LOC) Partitions
3.6
PSSA
Determine Item Probability Allocations, Validate Arch & Safety Assumptions
3.7
Validate System Level Requirements and Architecture
3.8
Allocate System Level Functions and Requirements to Items(s)
4.0
Implement HI LOC Items using DO-178 & DO-254
4.0
Implement LO LOC Items
Define RTA Monitoring Mechanisms
System / Item Development Activity
4761 Safety Activity

Figure A-1 Application Case Study Presentation Flow

A2.0 Notional Aircraft and Systems Concept Description

The ATCorp Team developed this notional aircraft and example case study framework for a new and novel DEP VTOL based on Uber Elevate’s eVTOL Common Reference Model (eCRM). This example aircraft and system definitions are further expanded in this appendix to support the needs of the Alternate Assurance Concept Application Case Study.

The Case Study DEP aircraft is a powered-lift vehicle to be developed and certificated under Normal, Utility, Acrobatic and Commuter Category Airplane regulations for VFR day use. The aircraft will be referred to as the eCRM-001 airplane and will have the key characteristics identified in Table A-1.

Table A-1 Notional Airplane Key Characteristics

Parameter	Notional Aircraft Characteristic
Crew	1
Passenger	4
Payload	900 lbs (408 kg)
Takeoff Gross Weight (TOGW)	5,130 lbs (2327 kg)
Range	60 miles (97 km)
Cruise Speed	130 kts (150 mph)
Stall Speed (airplane configuration)	80 kts (92 mph)
Operation	Visual Flight Rules (VFR)
Average Flight Duration	30 minutes

Figure A-2 presents eCRM-001 Notional DEP study airplane concept. The eCRM-001 is sized for a single pilot and four passengers in a conventional two by two seating arrangement. The aft row is raised for a stadium seating arrangement. The vehicle locates the center of gravity (CG) and center of thrust near the aft passenger row and features a lifting tail to dynamically control the aerodynamic center. Batteries are located in the wings to help with span loading weight as well as wing root bending moments.

The eCRM-001 is equipped with a single conventional pilot control and display to facilitate certificated and type rated pilots' control of vehicle functionality.

The eCRM-001 is planned for use in urban air mobility-air taxi scenarios.

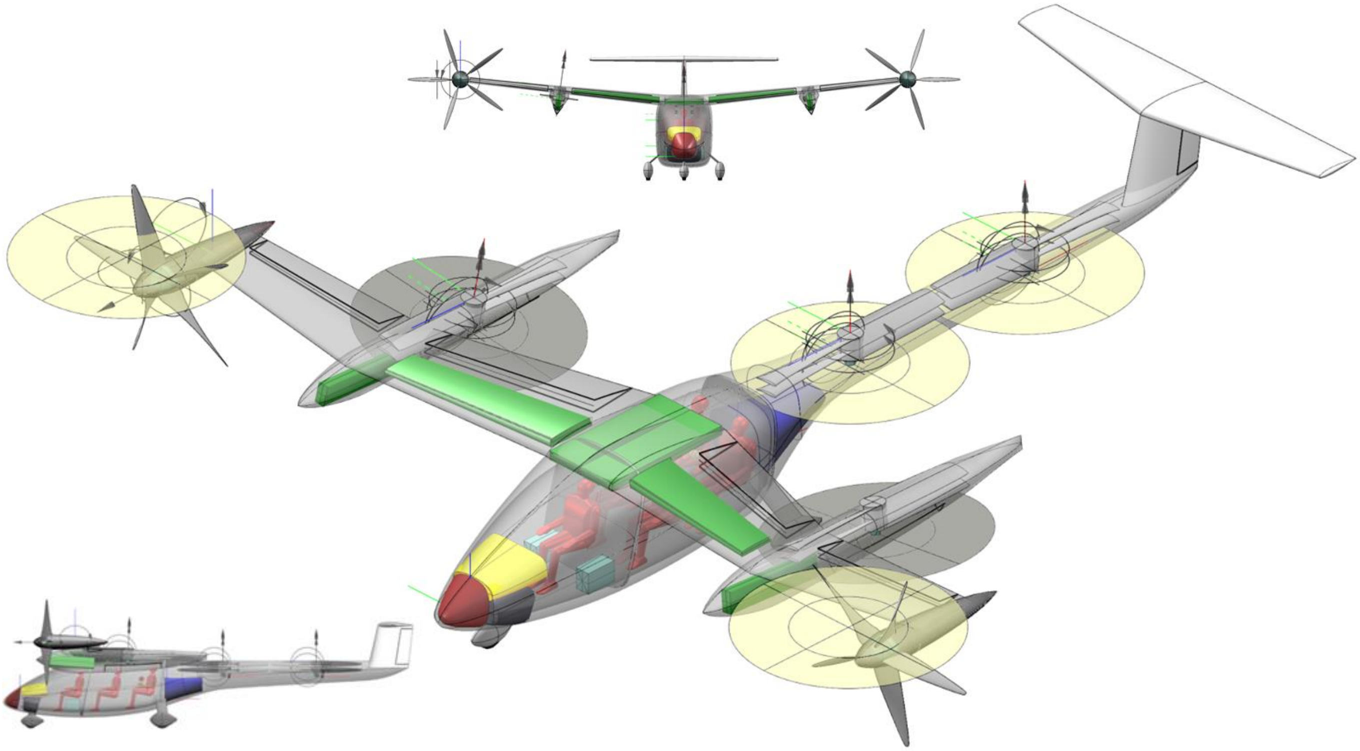


Figure A-2 eCRM-001 Notional DEP Study Vehicle

The eCRM-001 features a tilt rotor design augmented by multiple, retractable stacked lifting rotors for vertical flight. A conventional wing with a lifting T-tail is provided for forward flight. The set of six (6) stacked rotors concept provides a high lift – low noise eVTOL solution. Key configuration elements for the eCRM-001 are identified in Figure A-3.

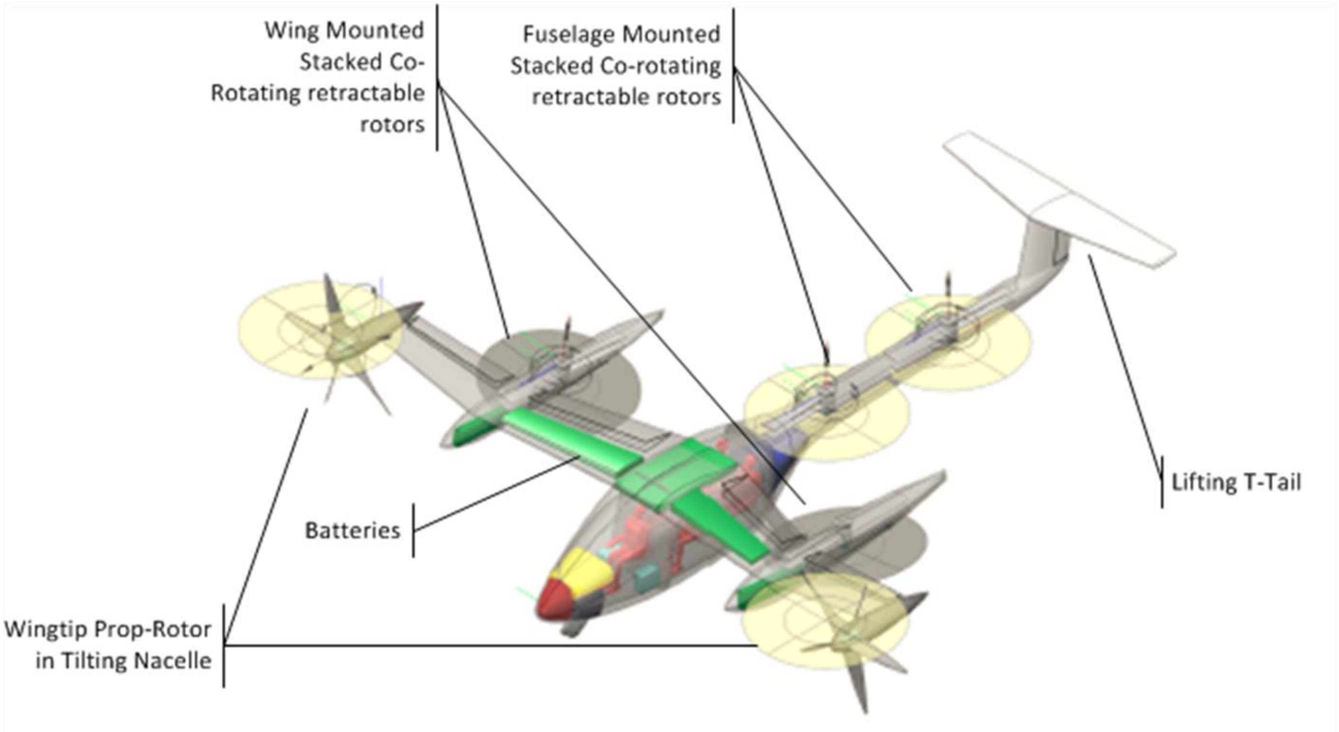

Wing Mounted Stacked Co-Rotating retractable rotors
Fuselage Mounted Stacked Co-rotating retractable rotors
Lifting T-Tail
Batteries
Wingtip Prop-Rotor in Tilting Nacelle

Figure A-3 eCRM-001 Key Configuration Elements

The eCRM-001 power train architecture is presented in Figure A-4. Each wing tip propeller is driven by independent dual motors with dual controllers integrated using sprag clutches. Six high voltage battery busses provide redundant electric power to the wingtip and stacked rotor effectors.

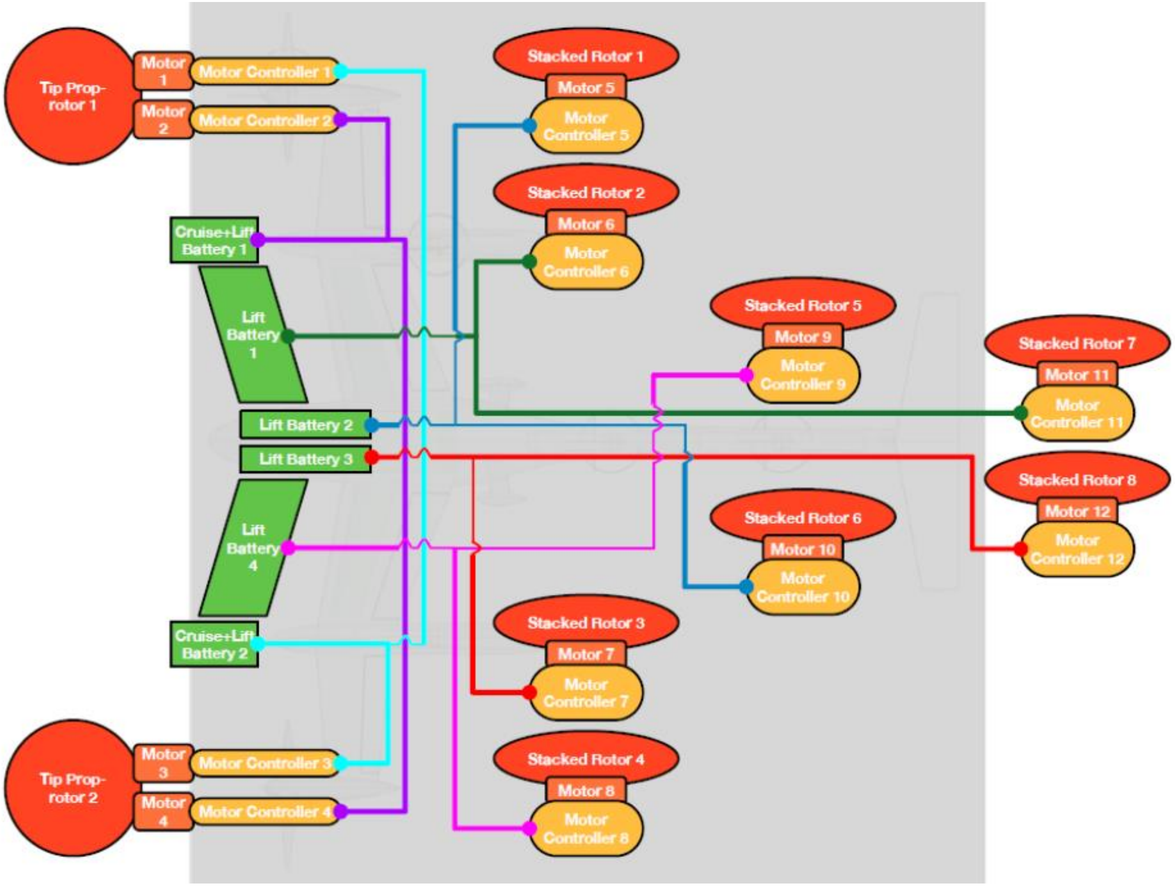


Figure A-4 eCRM-001 Powertrain Architecture

A duplex flight-propulsion control architecture, shown in Figure A-5, provides management of lift and flight control effectors to maintain continued safe flight and landing capabilities in the presence of implementation system failures.

The subsequent sections of the appendix will further detail a portion of the overall aircraft functionality so as to create the airplane data necessary to perform the research assurance comparison objectives.

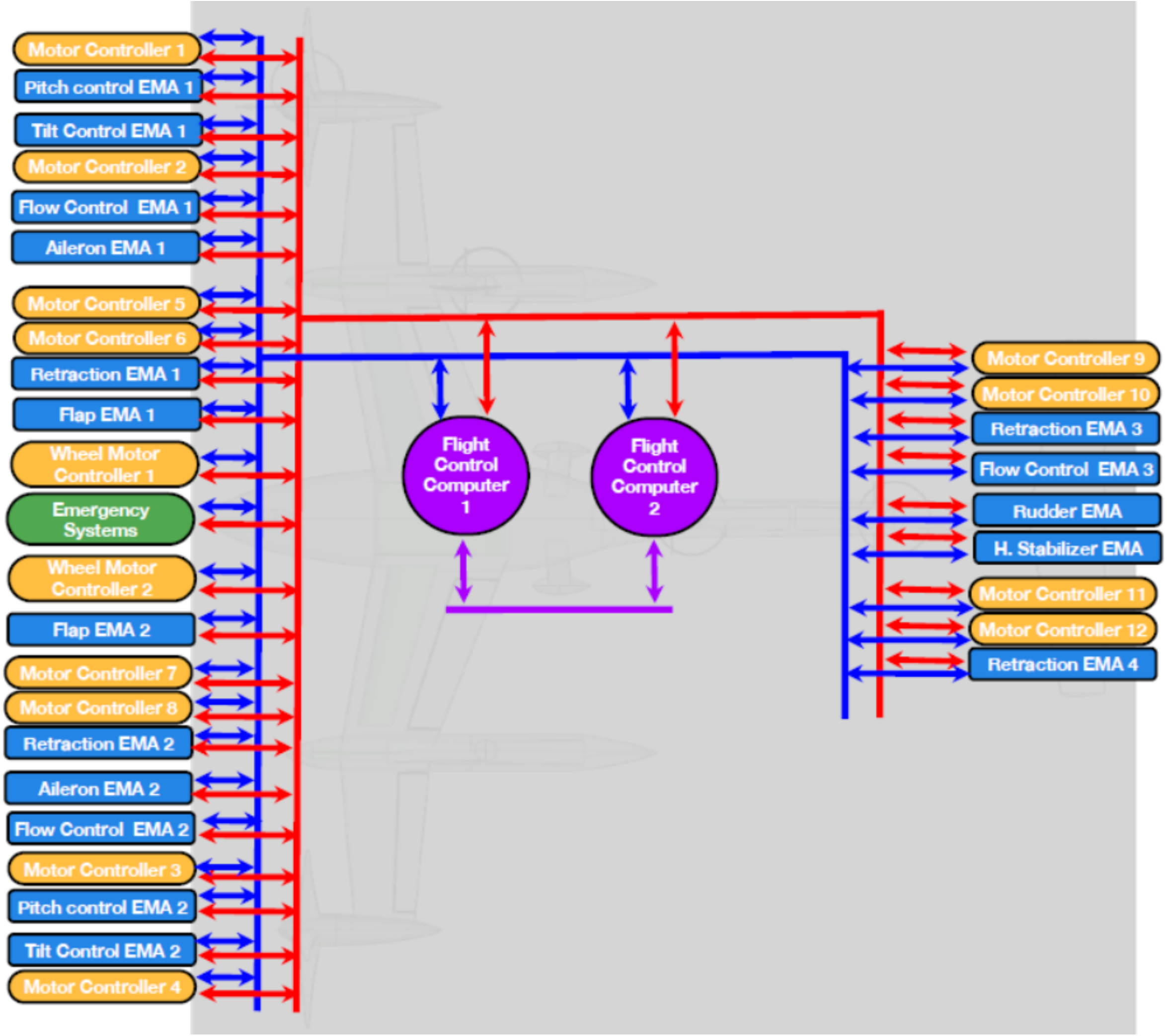


Figure A-5 Notional Flight-Propulsion Control Architecture

A2.1 – Identify Airplane Level Functions

The high level airplane functions for the eCRM-001 are shown in Figure A-6. The high level eCRM-001 airplane functions include airplane structure, provide control of movement, provide power generation and distribution to support control of movement, and accommodate maintenance of the systems and human occupants. On a typical aircraft development project each of these high level airplane functions would be further decomposed into lower level definitions of functionality.

Airplane Function 2, “Provide Control of Movement” will be the subject for the developed case study herein. The selected airplane level function is further decomposed into four sub-functions: Provide Controlled Vertical Takeoff-Landing (VTOL) Movement, Provide Controlled Forward Flight Movement, Provide Controlled Ground Movement and Provide Operational Awareness.

The “Provide Controlled VTOL Flight Movement” and “Provide Controlled Forward Flight Movement” functionality will be further decomposed for this case study.

Note: *Provide Controlled Ground Movement will not be further expanded in the case study. The Provide Operational Awareness functionality will be included to the level necessary to complete the Controlled VTOL and Forward Flight Movement functions.*

A2.2 – Identify airplane level functional requirements

The airplane level functional requirements are derived at the appropriate level of abstraction consistent with the function list defined (see Figure A-6). For this example, four aircraft level functions have been identified:

1. Provide Structural Integrity
2. Provide Control of Movement
3. Provide Power Generation & Distribution
4. Provide Loading, Maintenance, Ground Handling and Occupant Accommodation.

On a normal development, requirements for each of these functions, with the exception of provide structure, would be captured and validated per the planned airplane system development process.

A2.2.1 – Identify eCRM-001 Certification Strategy – Baseline Process

The eCRM-001 project certification plan is presented in Figure A-7 Example Artifact 1: eCRM Certification Plan.

Baseline Certification Plan Discussion:

As noted in Example Artifact 1, industry consensus standards will be used to demonstrate the eCRM-001 complies with the Part 23 and other selected regulations. The following ASTM standards are identified as the acceptable means of compliance (AMOC) for 14CFR Part 23 [3]:

- F3264-17 (18 is current) Standard Specification for Normal Category Aeroplanes Certification [15],
- F3061-17 Standard Specification for Systems and Equipment in Small Aircraft [16],
- F3230-17 Standard Practice for Safety Assessment of Systems and Equipment in Small Aircraft [17],
- F3153-15 Standard Specification for Verification of Avionics Systems [18].

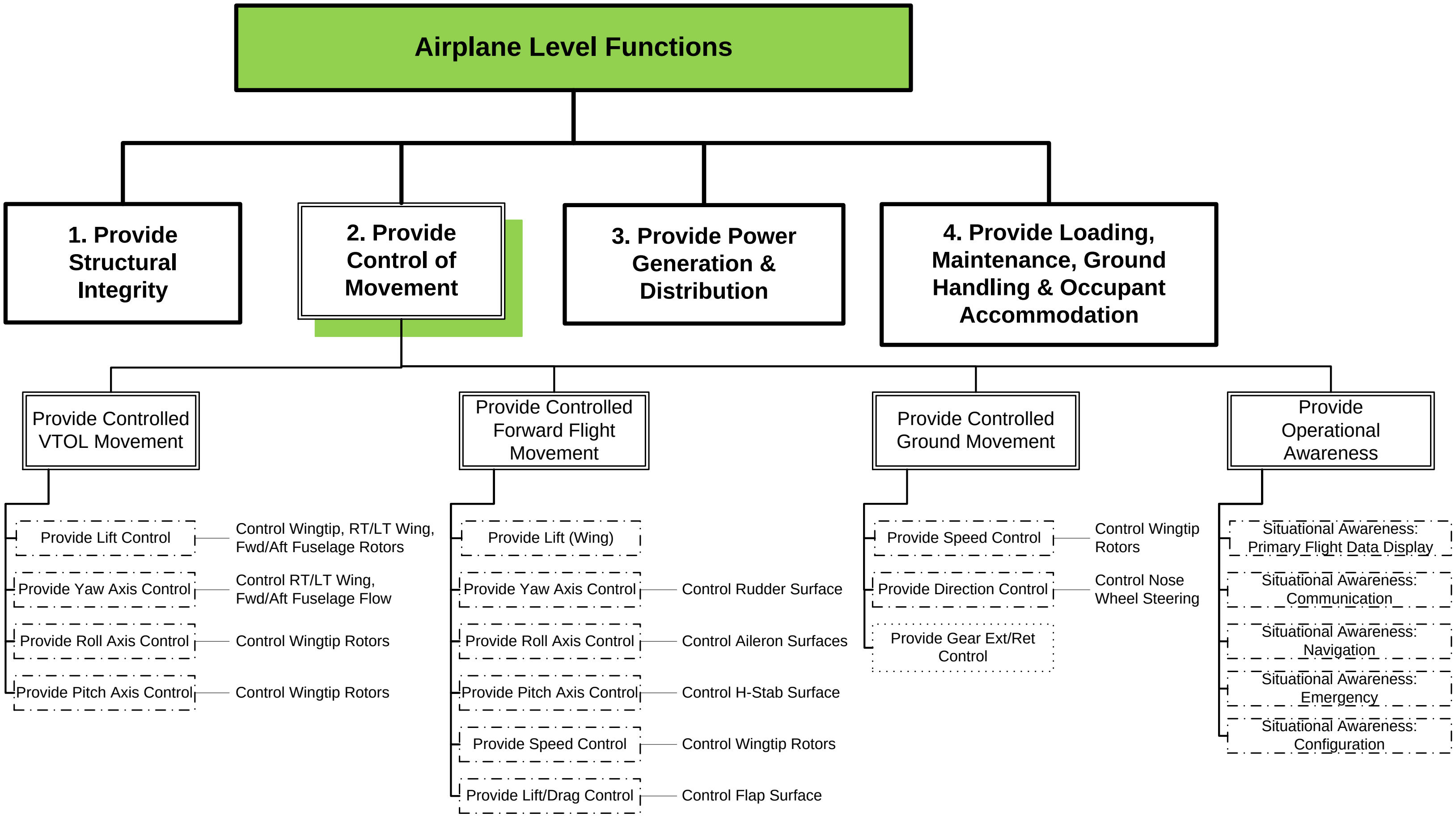

Airplane Level Functions
1. Provide Structural Integrity
2. Provide Control of Movement
3. Provide Power Generation & Distribution
4. Provide Loading, Maintenance, Ground Handling & Occupant Accommodation
Provide Controlled VTOL Movement
Provide Lift Control
Control Wingtip, RT/LT Wing, Fwd/Aft Fuselage Rotors
Provide Yaw Axis Control
Control RT/LT Wing, Fwd/Aft Fuselage Flow
Provide Roll Axis Control
Control Wingtip Rotors
Provide Pitch Axis Control
Control Wingtip Rotors
Provide Controlled Forward Flight Movement
Provide Lift (Wing)
Provide Yaw Axis Control
Control Rudder Surface
Provide Roll Axis Control
Control Aileron Surfaces
Provide Pitch Axis Control
Control H-Stab Surface
Provide Speed Control
Control Wingtip Rotors
Provide Lift/Drag Control
Control Flap Surface
Provide Controlled Ground Movement
Provide Speed Control
Control Wingtip Rotors
Provide Direction Control
Control Nose Wheel Steering
Provide Gear Ext/Ret Control
Provide Operational Awareness
Situational Awareness: Primary Flight Data Display
Situational Awareness: Communication
Situational Awareness: Navigation
Situational Awareness: Emergency
Situational Awareness: Configuration

Figure A-6 High Level eCRM-001 Airplane Functions

Figure A-7 Example Artifact 1: eCRM Certification Plan

BAART-ECRM-001-100

eCRM-001 Certification Plan

SIZE	FSCM NO	DWG NO	REV
A		CertPlan-100	-
SCALE		SHEET	1 of 8

REVISIONS

CN NO.	REV	DESCRIPTION	DATE	APPROVED
		Initial Release	01Oct2019	

1. Introduction

- ➢ New airplane type development project, aka eCRM-100
- ➢ Advance carbon composite construction
- ➢ Integrated electronic flight control – propulsion electronics (ATA 27 & 33)
- ➢ Advanced avionics flight deck featuring LCD "glass" displays & IMA mechanization

1.1. References

The following documents are referenced in this plan.

[1] 14CFR/CS Part 21, Amndmt 21-100, Certification Procedures for Products and Articles

[2] 14CFR/CS Part 23, Amndmt 23-64, Airworthiness Standards: Normal Category Airplanes

[3] 14CFR/CS Part 27, Amndmt 27- 50 Airworthiness Standards: Normal Category Rotorcraft

[4] AC 23.2010-1, dtd 03/27/17, FAA Accepted Means of Compliance Process for 14 CFR Part 23,

[5] No 92 Notice 23-81-NOA, Federal Register Volume 83, Accepted Means of Compliance; Airworthiness Standards: Normal Category Airplanes

[6] ASTM F3061- 17, Standard Specification for Systems and Equipment in Small Aircraft

[7] ASTM F3230-17, Safety Assessment of Systems and Equipment in Small Aircraft

[8] ARP4761A Guidelines and Methods for Conducting the Safety Assessment Process on Aircraft, Systems and Equipment

[9] eCRMAFHA-100, eCRM-001 Airplane Functional Hazard Assessment

Editor's Note: Document reference numbering within an example artifact will be to the documents listed as references in this section rather than the overall report reference list.

2. Certification Planning

The eCRM-001 will be certificated under 14CFR Part §21.17 using the applicable regulations from 14CFR Part 23 Amendment 23-64, Airworthiness Standards for Normal, Utility, Acrobatic and Commuter Category Airplanes and 14CFR Part 27 Amendment 27-35 Airworthiness Standards: Normal Category Rotorcraft. Industry Consensus Standards, as advised in AC 23.2010-1, "FAA Accepted Means of Compliance Process for 14 CFR Part 23", dtd 03/27/17, will be used to accomplish the safety intent of the regulations. The ASTM standards identified in the Federal Register "Accepted Means of Compliance; Airworthiness Standards: Normal Category Airplanes", Federal Register Volume 83, No 92 Notice 23-81-NOA) will form the basis for regulatory satisfaction.

Editor's Note: The airplane certification plan would normally capture compliance criteria planned for each regulation of the part. This example artifact captures only the regulations and compliance associated with safety assurance.

3. FHA Summary:

A summary of the eCRM-001 airplane level functions is presented in Figure CertPlan-1 Airplane Level Functions Diagram.

A summary of the catastrophic and hazardous functional failure conditions from reference [9] are summarized in Table CertPlan-1 eCRM-001 Airplane FHA Summary.

Table CertPlan-1 eCRM-001 Airplane FHA Summary

1	2	3	4	5
Function	**FC #**	**Failure Condition (Hazard Description)**	**Flight Phase**	**Classification**
Provide Control of Movement: Provide VTOL Movement Control	2.1.1.TL	Loss Vertical Lift MAC	T, L	Catastrophic
	2.1.2.TL	Loss of Yaw MAC	T, L	Hazardous
	2.1.3.TL	Loss of Roll MAC	T, L	Catastrophic
	2.1.4.TL	Loss of Pitch MAC	T, L	Catastrophic
	Etc.			
Provide Control of Movement: Provide Forward Flight Movement Control	2.2.2.TL	Loss of Roll MAC	F	Catastrophic
	2.2.3.TL	Loss of Pitch MAC	F	Catastrophic
	Etc.			

4. Safety Objectives and Assurance Levels:

- The safety assessment process will follow the activities outlined in ASTM F3230-17 using the methods and tools described in ARP4761.
- Safety objectives will be identified from safety assessments and prior experience.
- Development assurance levels will be assigned as recommended in ASTM F3061-17.

5. Novel or Unique Design Features:

- Tilt rotor augmented by fixed lift rotors for VTOL flight
- All electric propulsion using integrated flight and propulsion electronics based control system.

6. Certification Basis:

The eCRM-001 will be certificated to 14CFR Part 23, Amendment 23-64 [1] regulations. Advisory material, equivalent levels of safety (ELOS) findings, issue papers and special conditions captured herein will augment the basic 14CFR Part 23 regulatory requirement set.

The eCRM-001 Means of Compliance (MoC) Matrix is presented in Table eCRM-001 14CFR Part 23 Means of Compliance (MoC) Matrix. This matrix provides a summary extract of regulation compliance and references all pertinent information to be used in regulation compliance.

Editor's Note: Certification regulation and compliance method cross reference tables capture the Part 23 regulation text and planned compliance method(s). This Table eCRM-001 14CFR Part 23 Means of Compliance (MoC) Matrix example is limited to showing compliance to the regulations, §23.2500, §23.2505 and §23.2510 which use assurance methods as a primary means of compliance.

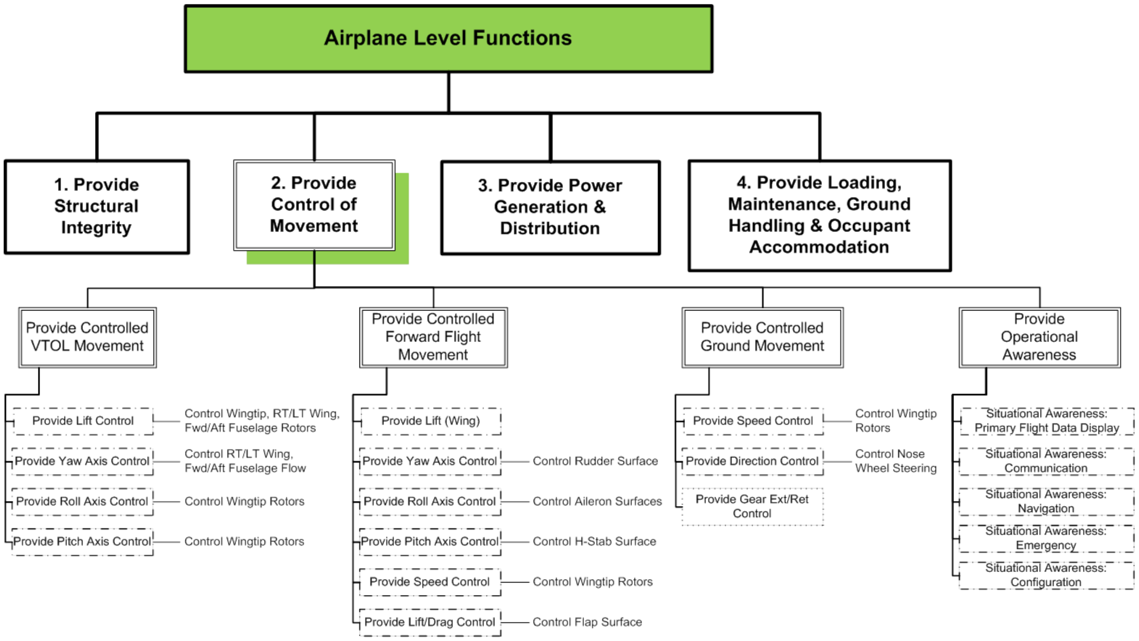

Airplane Level Functions
1. Provide Structural Integrity
2. Provide Control of Movement
3. Provide Power Generation & Distribution
4. Provide Loading, Maintenance, Ground Handling & Occupant Accommodation
Provide Controlled VTOL Movement
Provide Lift Control
Control Wingtip, RT/LT Wing, Fwd/Aft Fuselage Rotors
Provide Yaw Axis Control
Control RT/LT Wing, Fwd/Aft Fuselage Flow
Provide Roll Axis Control
Control Wingtip Rotors
Provide Pitch Axis Control
Control Wingtip Rotors
Provide Controlled Forward Flight Movement
Provide Lift (Wing)
Provide Yaw Axis Control
Control Rudder Surface
Provide Roll Axis Control
Control Aileron Surfaces
Provide Pitch Axis Control
Control H-Stab Surface
Provide Speed Control
Control Wingtip Rotors
Provide Lift/Drag Control
Control Flap Surface
Provide Controlled Ground Movement
Provide Speed Control
Control Wingtip Rotors
Provide Direction Control
Control Nose Wheel Steering
Provide Gear Ext/Ret Control
Provide Operational Awareness
Situational Awareness: Primary Flight Data Display
Situational Awareness: Communication
Situational Awareness: Navigation
Situational Awareness: Emergency
Situational Awareness: Configuration

Figure CertPlan-1 Airplane Level Functions Diagram

7. Compliance Methods:

The following methods will be used as identified in Table CertPlan-2 eCRM-001 14CFR Part 23 Means of Compliance Matrix columns 3 thru 8 using the design, inspection, analysis/similarity, and test (component, airplane ground or airborne) methods identified in column 2. Planned compliance reference information, associated with each regulation and/or regulation paragraph/clause is captured in column 9.

Table CertPlan-2 eCRM-001 14CFR Part 23 Means of Compliance (MoC) Matrix (subset extract)

1	2	3	4	5	6	7	8	9	10
eCRM-01 Airplane **FAA Project No. TC12345**	**An 'X' denotes compliance by the listed method(s):** **D = Design C = Component Test** **I = Inspection G = Ground Test** **A = Analysis/Similarity A = Airborne Test**	**Method of Compliance**							
					Test				
14 CFR 23 Paragraph	**Comment**	**D**	**I**	**A**	**C**	**G**	**A**	**Planned Compliance Reference**	**Actual MOC if Deviation from Plan**
Subpart F - Equipment									
23.2500 Airplane level systems requirements									
This section applies generally to installed equipment and systems unless a section of this part imposes requirements for a specific piece of equipment, system, or systems.								ASTM F3264-17, section 9.1 ASTM F3061-17	
(a) The equipment and systems required for an airplane to operate safely in the kinds of operations for which certification is requested (Day VFR, Night VFR, IFR) must be designed and installed to— (1) Meet the level of safety applicable to the certification and performance level of the airplane; and	The eCRM-001 will be certificated to operate safely in DAY VFR flight conditions.	X	X	X		X	X	ASTM F3061-17	
(2) Perform their intended function throughout the operating and environmental limits for which the airplane is certificated.		X		X		X	X	ASTM F3061-17, ASTM F3153-15	
(b) The systems and equipment not covered by paragraph (a), considered separately and in relation to other systems, must be designed and installed so their operation does not have an adverse effect on the airplane or its occupants.				X				ASTM F3061-17, ASTM F3153-15 ASTM F3230-17	

Table CertPlan-2 eCRM-001 14CFR Part 23 Means of Compliance (MoC) Matrix (subset extract)

1	2	3	4	5	6	7	8	9	10
eCRM-01 Airplane **FAA Project No. TC12345**	**An 'X' denotes compliance by the listed method(s):** **D = Design C = Component Test** **I = Inspection G = Ground Test** **A = Analysis/Similarity A = Airborne Test**	**Method of Compliance**							
					Test				
14 CFR 23 Paragraph	**Comment**	**D**	**I**	**A**	**C**	**G**	**A**	**Planned Compliance Reference**	**Actual MOC if Deviation from Plan**
23.2505 Function and installation.									
When installed, each item of equipment must function as intended.	Flight-Propulsion system operation will be described in a System Description Document and demonstrated to be appropriate for its intended function through inspection, analysis and test. Development process will be also be used to ensure intended function operation.	X	X	X	X	X	X	ASTM F3264-17, section 9.2 AC20-115C – Airborne Software Assurance AC20-152 - DO-254, Design Assurance Guidance for Airborne Electronic Hardware ARP4754A – Guidelines for Development of Civil Aircraft and Systems DO-178C – Software considerations in Airborne Systems & Equipment Certification DO-254 – Design Assurance Guidance for Airborne Electronic Hardware	

Table CertPlan-2 eCRM-001 14CFR Part 23 Means of Compliance (MoC) Matrix (subset extract)

1	2	3	4	5	6	7	8	9	10
eCRM-01 Airplane **FAA Project No. TC12345**	**An 'X' denotes compliance by the listed method(s):** **D = Design** **C = Component Test** **I = Inspection** **G = Ground Test** **A = Analysis/Similarity** **A = Airborne Test**	**Method of Compliance**							
					Test				
14 CFR 23 Paragraph	**Comment**	**D**	**I**	**A**	**C**	**G**	**A**	**Planned Compliance Reference**	**Actual MOC if Deviation from Plan**
23.2510 Equipment, systems and installations.									
For any airplane system or equipment whose failure or abnormal operation has not been specifically addressed by another requirement in this part, the applicant must design and install each system and equipment, such that there is a logical and acceptable inverse relationship between the average probability and the severity of failure conditions to the extent that:	A safety evaluation of the completed implementation using industry standards and methods will be used to show compliance.			X				ASTM F3264-17, section 9.3 ASTM F3230-17	
(a) Each catastrophic failure condition is extremely improbable;	A safety evaluation of the completed implementation using industry standards and methods will be used to show compliance.	X		X				ASTM F3230-17 ARP4761 – Guidelines & Methods for Conducting the Safety Assessment on Civil Airborne Systems & Equipment	
(b) Each hazardous failure condition is extremely remote; and	A safety evaluation of the completed implementation using industry standards and methods will be used to show compliance.	X		X				ASTM F3230-17 ARP4761 – Guidelines & Methods for Conducting the Safety Assessment on Civil Airborne Systems & Equipment	

Table CertPlan-2 eCRM-001 14CFR Part 23 Means of Compliance (MoC) Matrix (subset extract)

1	2	3	4	5	6	7	8	9	10
eCRM-01 Airplane FAA Project No. TC12345	An 'X' denotes compliance by the listed method(s): D = Design C = Component Test I = Inspection G = Ground Test A = Analysis/Similarity A = Airborne Test	Method of Compliance							
					Test				
14 CFR 23 Paragraph	Comment	D	I	A	C	G	A	Planned Compliance Reference	Actual MOC if Deviation from Plan
(c) Each major failure condition is remote.	A safety evaluation of the completed implementation using industry standards and methods will be used to show compliance.	X		X				ASTM F3230-17 ARP4761 – Guidelines & Methods for Conducting the Safety Assessment on Civil Airborne Systems & Equipment	

--------End of Airplane Certification Plan Excerpt-------

A2.3 Identify Airplane Certification Strategy – Alternate Assurance Concept

The application of the RTA alternate assurance approach does not require any substantive revision to the planned certification strategy for the eCRM-001 airplane. The same compliance criteria will be applied with only narrative revisions in CertPlan-100 section 4 added to discuss unique safety assurance approach.

A2.3.1 Identify RTA Goals and Airplane Level Requirements

This study example focuses on a general run-time assurance pattern, in which a highly automated control mode is coupled with a lower automation mode. The RTA goal is to reduce the time and expense associated with validating the design and verifying the implementation (V&V) of the high-automation algorithm by adding an RTA monitor and switch that prevents the high-automation algorithm from violating key safety requirements of the system. The intent is that validation and verification of the RTA components is easier than V&V of the high-automation algorithm.

This formulation is motivated by an assumption that urban air mobility (UAM) providers will initially deploy aircraft operated by highly trained pilots who have significant vertical-lift experience. Over time, however, these companies will transition to operations by minimally trained pilots who have no significant vertical lift experience. This will require a corresponding increase in the level of flight control automation, permitting these lesser trained pilots to handle the complexity of hover and transition flight modes associated with UAM operations. This increased automation will require certification of flight control software that is significantly different than previous Part 23 designs, and it is expected that the engineering expertise to directly produce and evaluate high-DAL implementations of this software will be lacking.

The goal of this RTA example application is to permit the gradual roll out and subsequent maturation of these higher-level control algorithms during the initial UAM business phase, while the company is employing highly trained pilots for its operations. The concept involves placing both low-automation and high-automation software on the aircraft, together with RTA monitoring and switching components. The low-automation algorithms will constitute the high-confidence system; from a certification perspective, this is the primary system. Its design, implementation, and approval processes will follow standard practices for a high DAL system, which will be based on an assumption that a highly trained pilot is operating the aircraft. The high-automation software will constitute the low-confidence system; it will be assigned a low DAL.

During flight, while the high-automation algorithms are operating, the RTA system will monitor the aircraft state for any impending violation of safety requirements. When necessary, it will switch to the low-automation software to prevent such violations. Because the pilots are highly trained, this switch in control could be reasonable, provided that the pilots are familiar with this switching behavior and such a switch is properly announced. The RTA system will be assigned a high DAL, commensurate with that of the low-automation control algorithms.

This architectural solution will allow UAM companies to gradually improve their high-automation algorithms as they gain experience through real-world operations, without the need to re-certify each new version of those algorithms. This will significantly lower the development activities and reduce the time necessary to deploy updated control algorithms. It will also provide an opportunity to collect large quantities of real-world data on the behavior of the high-automation algorithms, which can be leveraged if/when they are later certified as primary systems for use by lesser trained pilots.

Examples of this RTA Pattern

Due to the large number of rotating elements, control surfaces, and other actuators, it is unlikely that even a highly trained pilot will be capable of flying the aircraft in a “stick-to-surface” operational mode. A more reasonable control approach would be to have the pilot use the stick (or other inceptor) to command desired roll and pitch angles (called Attitude Command/Attitude Hold, or ACAH), with feedback control and an allocator designed to achieve those commands. With sufficient training, pilots should be capable of accurate aviation using this approach. This control approach could be considered low-level automation, and the team believes that it could be designed and implemented using a traditional assurance approach.

However, studies have suggested that translation-rate control (TRC), in which the pilot uses the stick to command a horizontal velocity vector rather than an attitude, may be a better approach for pilots with lesser training. (For example, see [105] and the related publications of the myCopter project.) This control architecture adds a layer of complexity on top of a basic ACAH controller, and it would constitute a higher level of automation. The team is not aware of any commercial aircraft that have been certified and that employ this control approach. In this case, the time and effort required for V&V of a TRC algorithm in this application will be significant. However, if operated by a skilled pilot, trained to safely fly the aircraft in ACAH mode, the TRC mode could be paired with a separate ACAH mode and RTA components that switch control modes as appropriate. This could significantly reduce the burden of V&V for the TRC algorithm at the cost of V&V for the RTA components. The team asserts that this could be a reasonable approach because analysis activities for the RTA components can largely overlap those of the ACAH algorithm, and they are much less onerous than V&V of the TRC algorithms they guard.

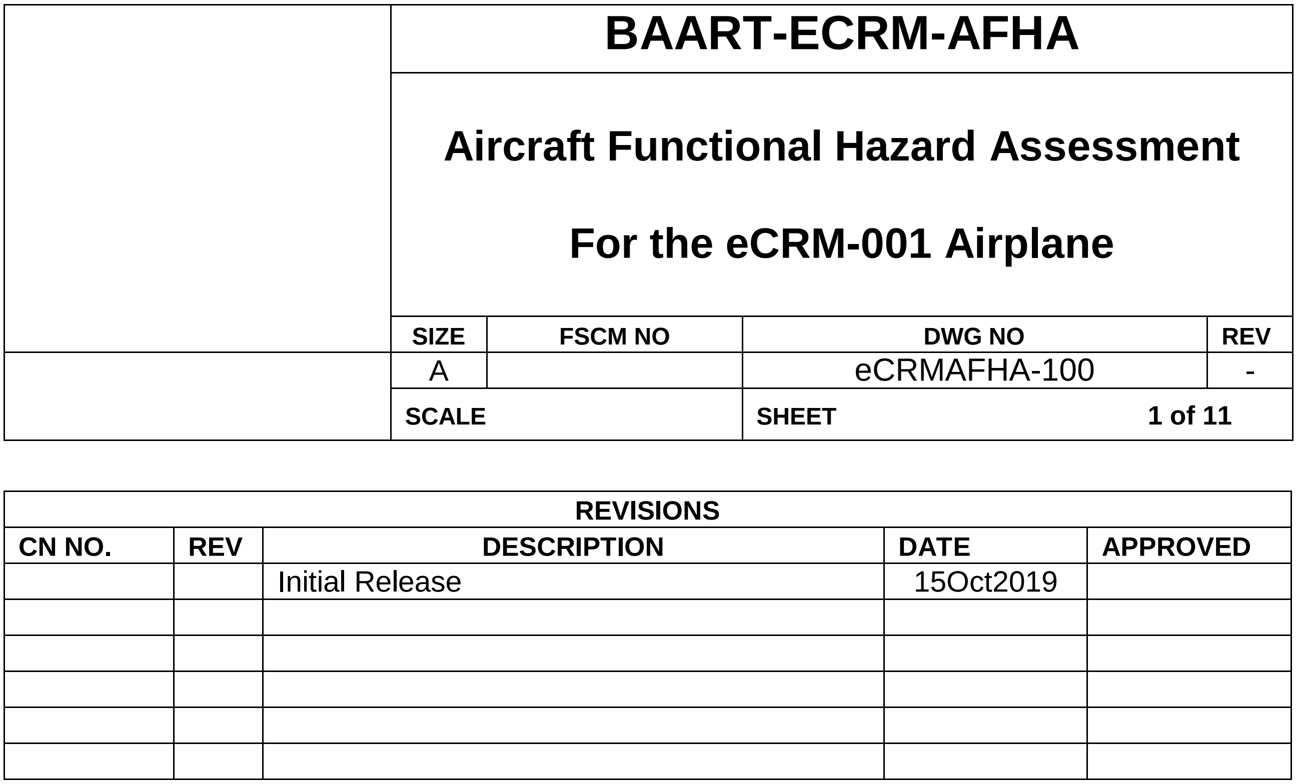
This basic pattern can be extended to higher levels of automation. Hovering flight mode, employed during the initial and final flight segments, requires a high level of energy expenditure because the vast majority of lift is produced directly from rotating surfaces of the aircraft. It is desirable to minimize the amount of time spent in hover mode to conserve electrical energy and maximize the number of passenger carrying flights before a full recharge cycle is needed. An autoland control algorithm could be developed specifically for this purpose. Autoland control algorithms for UAM applications are expected to be highly complex, with V&V-associated costs running much higher than those for TRC and ACAH algorithms. However, if the aircraft has a certified TRC implementation and a pilot trained to land the aircraft using that control mode, RTA monitoring and switching components could be designed to switch from autoland control mode to TRC mode preventing the violation of any safety requirements.

In this study, we look at the autoland scenario in more detail.

A2.4 Identify Airplane Level Safety Objectives (AFHA)

The airplane level safety objectives are established by an analysis of the planned functions to be implemented using a functional hazard assessment (FHA) methodology described in ARP4761 [23]. An extract of the eCRM-001 FHA is provided in Figure A-8 Example Artifact 2, Aircraft Functional Hazard Assessment For the eCRM-001 Airplane.

Figure A-8 Example Artifact 2: eCRM AFHA



	BAART-ECRM-AFHA			
	Aircraft Functional Hazard Assessment **For the eCRM-001 Airplane**			
	SIZE	**FSCM NO**	**DWG NO**	**REV**
	A		eCRMAFHA-100	-
	SCALE		**SHEET** **1 of 11**	

REVISIONS				
CN NO.	**REV**	**DESCRIPTION**	**DATE**	**APPROVED**
		Initial Release	15Oct2019	

1. Introduction

The assessment captured herein represents the airplane level functional hazard assessment (AFHA) for the eCRM-001 VTOL airplane.

1.1. References

The following documents are referenced herein.

[1] eCRM-001 Airplane Design Requirements Document

[2] 14CFR/CS Part 23, Amendment 23-64

[3] ASTM F3230-17, "Safety Assessment of Systems and Equipment in Small Aircraft"

[4] ASTM F3061-17, "Standard Specification for Systems and Equipment in Small Aircraft"

[5] ARP4761 "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Aircraft Systems and Equipment"

Editor's Note: Document reference numbering within an example artifact will be to the documents listed as references in this section rather than the overall report reference list.

1.2. Glossary

This section captures specific terms and definitions used within the AFHA.

Term	Definition
Uncommanded	Activation of a function without pilot command input or erroneously activated due to equipment failure.
Minimum Acceptable Control (MAC)	An aircraft configuration under which the normal acceptable control performance criteria will still be satisfied and when lost will result in the failure condition effects described.

2. Airplane Description Summary

Editor's note: Duplicate airplane description summary removed for brevity.

3. AFHA Development

The AFHA process accomplished herein is in accordance with ARP4761 [5] recommended guidelines.

3.1. AFHA Inputs

The airplane development process identified the airplane level functions captured in Table AFHA-1 eCRM-001 Airplane Function List. These functions will be the subject of the safety evaluation herein.

3.1.1. Review & Confirm Airplane Functions

Editors' Note: Not included in this example for brevity.

Table AFHA-1 eCRM-001 Airplane Function List

1. Provide structural integrity	2.4.3 Navigation Awareness
2. Provide Control of Movement	2.4.4 Emergency Awareness
2.1 Provide VTOL Movement Control	2.4.5 Configuration Awareness
2.2 Provide Forward Flight Movement Control	3. Provide Power Generation & Distribution
2.3 Provide Ground Movement Control	4. Provide Loading, Maintenance, Ground Handling & Occupant Accommodation
2.4 Provide Situational Awareness	4.1 Provide breathable atmosphere
2.4.1 Primary Flight Awareness	4.1.1 Provide oxygenated atmosphere
2.4.1.1 Display attitude	4.1.2 Prevent atmosphere toxicity
2.4.1.2 Display altitude	4.1.3 Provide controlled temperature
2.4.1.3 Display airspeed	4.2 Provide cabin temperature control
2.4.2 Communication Awareness	

3.2. Determine Failure Conditions

Editors' Note: Only the "Provide VTOL Movement Control, function 2.1, is developed in this example. All other airplane functions, at the level of the functional breakdown defined in 4.1, would be developed in a similar fashion.)

3.2.1. Failure Condition Identification Matrix

A failure condition identification matrix was constructed for the function "*Provide VTOL Movement Control*". This initial matrix is presented in Table AFHA-2 eCRM-001 Failure Condition Identification Matrix. Postulated failure condition descriptions are captured for Total Loss of function, Partial Loss of Function and Malfunction (erroneous operation) of Function.

Table AFHA-2 eCRM-001 Failure Condition Identification Matrix

ID #	Aircraft Function	Total Loss	Partial Loss	Malfunction
2	Provide Control of Movement			
2.1	Provide VTOL Movement Control			
2.1.1	Control lift for vertical movement	**2.1.1.TL** Loss of ability to control lift for vertical movement	**2.1.1.PL** Partial loss of ability to control lift for vertical movement	**2.1.1.MF1** Uncommanded lift **2.1.1.MF2** Erroneous lift intensity – excessive **2.1.1.MF3** Erroneous lift intensity - diminished
2.1.2	Control Yaw during vertical movement			
2.1.3	Control Roll during vertical movement			
2.1.4	Control Pitch during vertical movement			
2.2	Provide Forward Flight Movement Control			

3.2.2. Pilot Awareness

Editors' Note: Not included in this example for brevity.

3.3. Assess Failure Condition Effects

The effects of each of the identified failure condition on the aircraft, flight crew and occupants other than the flight crew have been assessed. The effects are captured based on their immediate effect on aircraft, flight crew and occupants during the phase of flight being analyzed.

The captured effects of each failure condition are shown in column (5) of the AFHA worksheet tables.

3.3.1. eCRM-001 Flight Profile

The eCRM-001 aircraft flight of average duration is presented in Figure AFHA-1 eCRM-001 Average Flight Profile. The nominal flight is divided into four flight phase groups, with individual flight phase durations as presented in Table AFHA-3 eCRM-001 Airplane Flight Phases.

Editor's Note: Not all eCRM-001 operational flight phases have been included in the study example system definition.

Table AFHA-3 eCRM-001 Airplane Flight Phases

Flight Phase	Flight Time	Flight Phase Code	Flight Phase
Ground	68s	G1	Taxi Out
	68s		Taxi In
Takeoff	18s	T1	Break ground to Hover
	60s	T2	Transition – Hover to Forward Flight
Forward Flight	91s	F1	Climb
	1212s	F2	Cruise
	91s	F3	Descent
		F4	Go Around
Landing	86s	L1	Transition – Forward Flight to Hover
	40s	L2	Hover – Descend to ground
	1734s 30 min	ALL	All flight phases

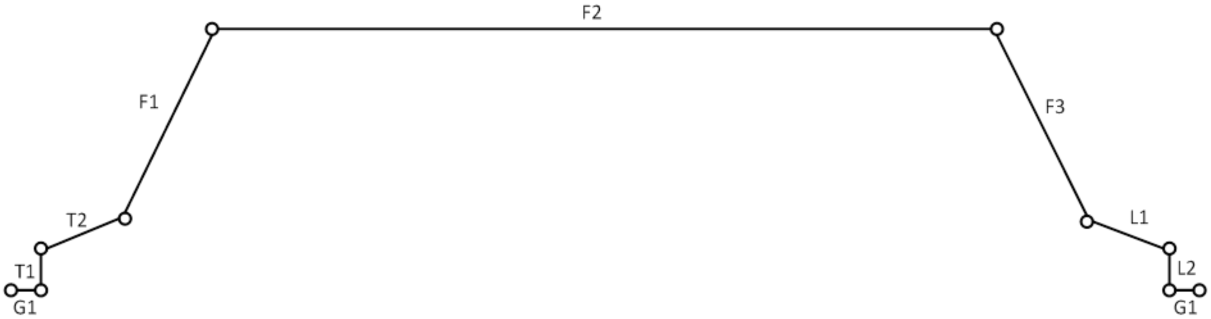


Figure AFHA-1 eCRM-001 Average Flight Profile

3.3.2. Operational Conditions

Editors' Note: Not included in this example for brevity.

3.3.3. Environmental Conditions

Editors' Note: Not included in this example for brevity.

3.4. Classify Failure Conditions Based on Effect Severity

Each failure condition has been classified based on its effects by applying the qualitative classification criteria provided in Reference [3], as applicable to this type of airplane. The failure condition classifications presented in Reference [3] are reproduced in Table AFHA-4 F3230-17 Failure Condition Classifications for convenience.

Table AFHA-4 F3230-17 Failure Condition Classifications

FC Classification based on Effect Area	**Negligible**	**Minor**	**Major**	**Hazardous**	**Catastrophic**
Effect on Aircraft	No effect on operational capabilities or safety	Slight reduction in functional capabilities or safety margins	Significant reduction in functional capabilities or safety margins	Large reduction in functional capabilities or safety margins	Normally with hull loss
Effect on Flight Crew	No effect on flight crew	Slight increase in workload or use of emergency procedures	Physical discomfort or a significant increase in workload	Physical distress or excessive workload impairs ability to perform tasks	Fatal injury or incapacitation
Effect on Occupants	Inconvenience for passengers	Physical discomfort for passengers	Physical distress to passengers, possibly including injuries	Serious or fatal injury to an occupant	Multiple fatalities
Allowable Qualitative Probability (F3230-17 [7], Table 4)	No Probability Requirement	Probable	Remote	Extremely Remote	Extremely Improbable
Allowable Quantitative Probability (F3230-17 [7], Table 5)	No Probability Requirement	AW-I ≤ 10^{-3} AW II ≤ 10^{-3}	AW-I ≤ 10^{-4} AW II ≤ 10^{-5}	AW-I ≤ 10^{-5} AW II ≤ 10^{-6}	AW-I ≤ 10^{-6} AW II ≤ 10^{-7}

Note: AW – Airworthiness Level of F3230-17 Table 3 is based on Assessment Level assigned per F3061-17. For the eCRM-001, the Assessment Level is "II".

The classification of each failure condition for each flight phase is captured in Column (5) of the AFHA worksheet tables.

3.5. AFHA Assumptions and Hazard Classification Criteria

Assumptions made while accomplishing the effect evaluation of each failure condition have been captured and numerically identified for reference. Table AFHA-5 presents the analysis assumptions and notes that have been made during the development of this functional hazard assessment.

Table AFHA-5 AFHA Assumptions (ASMP)/Notes

Assumption Identifier	**Description**
ASMP 2.1.1-1	Minimum acceptable vertical lift capability during Takeoff or Landing flight phases at identified worst case conditions is provided by any of the following configurations: • 2 wingtip rotors, 2 wing lift rotor stacks, 1 FWD lift rotor, • 2 wingtip rotors, 2 wing lift rotor stacks, 1 AFT lift rotor, • 1 wingtip rotor, 2 wing lift rotor stacks, 1 FWD lift rotor, 1 AFT lift rotor, • 2 wingtip rotors, 1 wing lift rotor, 1 FWD lift rotor and 1 AFT lift rotor.
ASMP 2.1.1-2	Performance is as measured on “standard day”.
ASMP 2.1.2-1	Minimum acceptable directional (yaw) control capability during Takeoff or Landing flight phases at identified worst case conditions is provided by any of the following: • 1 wingtip rotor, 2 wing lift rotors, 1 FWD lift rotor, 1 AFT lift rotor, 1 wingtip tilt mechanism, 2 wing lift flow control, AFT flow control • 2 wingtip rotors, 1 wing lift rotor, 1 FWD lift rotor, 1 AFT lift rotor, 2 wingtip tilt mechanisms, 2 wing lift flow control, AFT flow control • 2 wingtip rotors, 2 wing lift rotors, 1 AFT lift rotor, 2 wingtip tilt mechanisms, 2 wing lift flow control, AFT flow control • 2 wingtip rotors, 2 wing lift rotors, 1 FWD lift rotor, 2 wingtip tilt mechanisms, 2 wing lift flow control, AFT flow control • 2 wingtip rotors, 2 wing lift rotors, 1 FWD lift rotor, 1 AFT lift rotor, 1 wingtip tilt mechanism, 2 wing lift flow control, AFT flow control • 2 wingtip rotors, 2 wing lift rotors, 1 FWD lift rotor, 1 AFT lift rotor, 2 wingtip tilt mechanisms, 1 wing lift flow control, AFT flow control • 2 wingtip rotors, 2 wing lift rotors, 1 FWD lift rotor, 1 AFT lift rotor, 2 wingtip tilt mechanisms, 2 wing lift flow control.
ASMP 2.1.3-1	Minimum acceptable lateral (roll) control capability during Takeoff or Landing flight phases at identified worst case conditions is provided by any of the following: • 1 wingtip rotor, 2 wing lift rotors, • 2 wingtip rotors, 1 wing lift rotor.

Table AFHA-5 AFHA Assumptions (ASMP)/Notes

Assumption Identifier	**Description**
ASMP 2.1.4-1	Minimum acceptable longitudinal (pitch) control capability during Takeoff or Landing flight phases at identified worst case conditions is provided by any of the following: • 2 wingtip rotors, 2 wing lift rotors, and 1 FWD lift rotor; • 2 wingtip rotors, 2 wing lift rotors, and 1 AFT lift rotor; • 1 wingtip rotor, 2 wing lift rotors, 1 FWD lift rotor and 1 AFT lift rotor; • 2 wingtip rotor, 1 wing lift rotors, 1 FWD lift rotor and 1 AFT lift rotor.
ASMP 2.2.1-1	Minimum acceptable vertical lift during Forward Flight phases is provided by the wing and tail structure.
Etc.	

4. AFHA Output Summary

This section summarizes the failure effects and their associated severity classifications. For details associated with each failure conditions see the populated worksheets in Table AFHA-7 eCRM-001 Airplane Functional Hazard Assessment Worksheets (excerpt).

Table AFHA-6 eCRM-001 Catastrophic & Hazardous FC Summary

1	2	3	4	6
Function	**FC #**	**Failure Condition (Hazard Description)**	**Flight Phase**	**Classification**
Provide Control of Movement: Provide VTOL Movement Control	2.1.1.TL	Loss Vertical Lift MAC	T, L	Catastrophic
	2.1.1.MF-1	Uncommanded lift	T, L	Catastrophic
	2.1.2.TL	Loss of Yaw MAC	T, L	Hazardous
	2.1.3.TL	Loss of Roll MAC	T, L	Catastrophic
	2.1.4.TL	Loss of Pitch MAC	T, L	Catastrophic
	Etc.			
Provide Control of Movement: Provide Forward Flight Movement Control	2.2.2.TL	Loss of Roll MAC	F	Catastrophic
	2.2.3.TL	Loss of Pitch MAC	F	Catastrophic
	Etc.			

Editor's Note: For an actual vehicle certification project, each of the AFHA failure conditions would be developed in their entirety. The example system definition herein is developed only to the level of detail necessary to support the goals of the study.

Table AFHA-7 eCRM-001 Airplane Functional Hazard Assessment Worksheets (excerpt)

FUNCTIONAL HAZARD ASSESSMENT						
Function:	**VTOL Movement Control**					**Rev Date:** **01 October 2019**
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
2.1.1.TL	Loss of minimum acceptable vertical lift control (Lift MAC)	T1, T2, T3, L1, L2	A) Airplane unable to provide continued safe flight along desired flight path. Airplane impact with ground or surroundings resulting in significant airplane damage or hull loss. B) Flight crew unable to maintain desired flight path. Flight Crew fatalities. C) Passenger fatalities.	I	ASA/SSA	ASMP 2.1.1-1
2.1.1.MF-1	Uncommanded vertical lift	T1, T2, T3, L1, L2	A) Airplane unstable along desired flight path. Unpredictable climb or descent may result in impact with ground or surroundings resulting in significant airplane damage or hull loss. B) Flight crew unable to maintain desired flight path. Potential Flight Crew fatalities. C) Passenger fatalities.	I	ASA/SSA	
2.1.1.MF-2	Erroneous vertical lift intensity – lift differs from commanded - excessive	T1, T2, T3, L1, L2	A) Airplane climb rate exceeds commanded values resulting in increased climb performance. B) Slight increase in Flight Crew workload to compensate for increased lift performance. C) No effect.	IV	ASA/SSA	

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

- CLASS I — CATASTROPHIC
- CLASS II HAZARDOUS
- CLASS III MAJOR
- CLASS IV MINOR
- CLASS V NO EFFECT

FUNCTIONAL HAZARD ASSESSMENT						
Function:	**VTOL Movement Control**					**Rev Date:** **01 October 2019**
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
2.1.1.MF-3	Erroneous vertical lift intensity – lift differs from commanded - diminished	T1, T2, T3, L1, L2	A) Airplane climb rate underperforms commanded values resulting in decreased climb performance. B) Significant increase in Flight Crew workload to compensate for decreased lift performance. C) No effect.	III	ASA/SSA	
2.1.2.TL	Loss of minimum acceptable yaw control (Yaw MAC)	T1, T2, T3, L1, L2	A) Airplane unable to provide continued directional control. Large reduction in safety margin. B) Crew experiences excessive workload to control direction resulting inability to perform required tasks. C) Potential injury or death to some of the passengers.	II	ASA/SSA	ASMP 2.1.2-1
2.1.3.TL	Loss of minimum acceptable roll control (Roll MAC)	T1, T2, T3, L1, L2	A) Airplane unable to provide continued safe flight along desired flight path. Airplane impact with ground or surroundings resulting in significant airplane damage or hull loss. B) Flight crew unable to maintain desired flight path. Flight crew fatalities. C) Passenger fatalities.	I	ASA/SSA	ASMP 2.1.3-1
2,1,4,TL	Loss of minimum acceptable pitch control (Pitch MAC)	T1, T2, T3, L1, L2	A) Airplane unable to provide continued safe flight along desired flight path. Airplane impact with ground or surroundings resulting in significant airplane damage or hull loss. B) Flight crew unable to maintain desired flight path. Flight crew fatalities. C) Passenger fatalities.	I	ASA/SSA	ASMP 2.1.4-1

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

•	CLASS I —	CATASTROPHIC
•	CLASS II	HAZARDOUS
•	CLASS III	MAJOR
•	CLASS IV	MINOR
•	CLASS V	NO EFFECT

--------End of Airplane FHA Excerpt-------

A2.5 Allocate Airplane Level Functions to Systems

The airplane level function “Provide Control of Movement” and sub-function “Provide Controlled VTOL Movement” are further decomposed for system definition. For the example, Provide Controlled VTOL Movement has been allocated into Provide Manual or Automated Control using the wingtip, wing and fuselage rotor effectors as shown in Figure A-9.

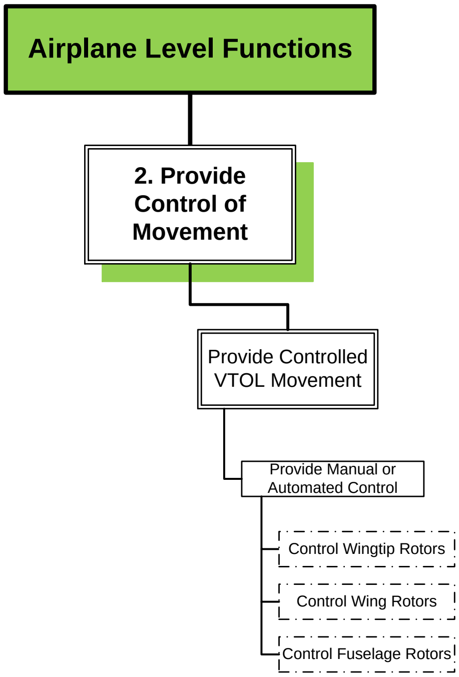

Airplane Level Functions
2. Provide Control of Movement
Provide Controlled VTOL Movement
Provide Manual or Automated Control
Control Wingtip Rotors
Control Wing Rotors
Control Fuselage Rotors

Figure A-9 Controlled VTOL Movement Decomposition

A high level diagram of this planned Flight and Propulsion Control capability is presented in Figure A-10 High Level Flight & Propulsion System Architecture. The various airplane control effectors in Figure A-10 are highlighted in blue. The high-level interfaces to the electrical power, electronics, sensors and pilot controls are also shown.

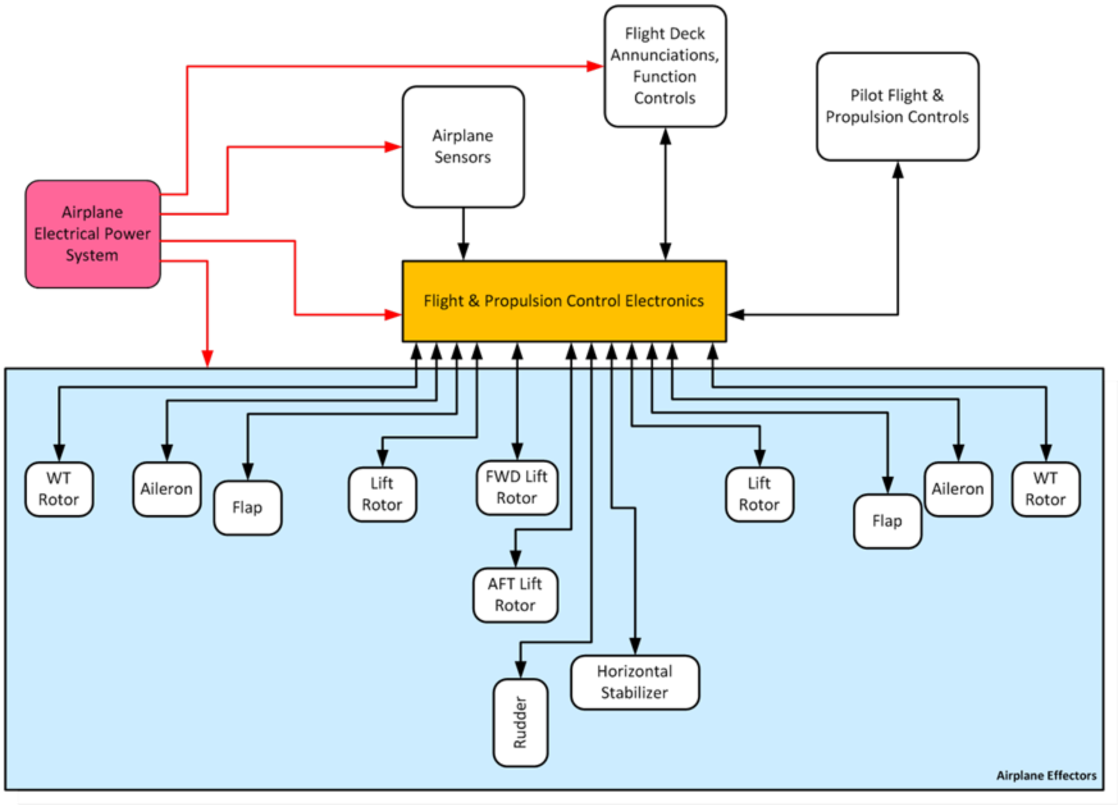

Flight Deck Annunciations, Function Controls
Pilot Flight & Propulsion Controls
Airplane Sensors
Airplane Electrical Power System
Flight & Propulsion Control Electronics
WT Rotor
Aileron
Flap
Lift Rotor
FWD Lift Rotor
AFT Lift Rotor
Rudder
Horizontal Stabilizer
Lift Rotor
Flap
Aileron
WT Rotor
Airplane Effectors

Figure A-10 High Level Flight & Propulsion System Architecture

The airplane system development process allocates the high level airplane functions to various planned airplane systems or specific system elements in one system. Table A-2 Control of VTOL Flight Movement Function Allocations presents an allocation of the various airplane control movement functions (both manual and automatic) to the planned airplane flight and propulsion system functional elements. Each of the eCRM-001 Provide Control of Movement functions is allocated to one or more of the planned flight and propulsion control capabilities.

The “Provide Operational Awareness” function contains the sensors that will be used in the control movement functions.

Table A-2 Control of VTOL Flight Movement Function Allocations

1	2	3	4	5	6	7	8	9	10
System Function	Wingtip Rotor Control	LT/RT Wing Lift Rotor Control	FWD/AFT Fuselage Lift Rotor Control	LT/RT Wing Lift Rotor Flow Control	FWD/AFT Fuselage Lift Rotor Flow Control	Rudder Surface Control	Aileron Surface Control	H-Stab Surface Control	Flap Surface Control
Control VTOL Flight:									
Control Vertical Flight	X	X	X						
Control Yaw during VTOL Flt				X	X				
Control Roll during VTOL Flt	X	X		X	X				
Control Pitch during VTOL Flt	X		X						
Control Forward Flight:									
Control Yaw during Forward Flt						X			
Control Roll during Forward Flt							X		
Control Pitch during Forward Flt								X	
Control Forward Speed	X								
Control Lift/Drag									X

Editor's Note: The pitch, roll and yaw nomenclature has been used to simplify the technical discussion. It is understood that this type of vehicle involves control with 6-degrees of controlled motion.

A2.6 Develop Aircraft Functional Level Architecture

The aircraft level Flight and Propulsion Control System (FPCS) architecture is now developed. The “notional” duplex architecture presented in section A-1 is refined in the context of the airplane functional allocations accomplished in A2.5.

A2.6.1 eCRM-001 Integrated Flight Propulsion Control System Description

Figure A-11 eCRM-001 Flight Propulsion Control Functional Block Diagram presents the example study system integrated flight-propulsion control capabilities that will be incorporated into the airframe. For system definition purposes, the functionality is organized into four groups: Central Controller, Wing Tip Control, Lift Control and Surface Control.

Redundant electric power is provided to each of the four groups.

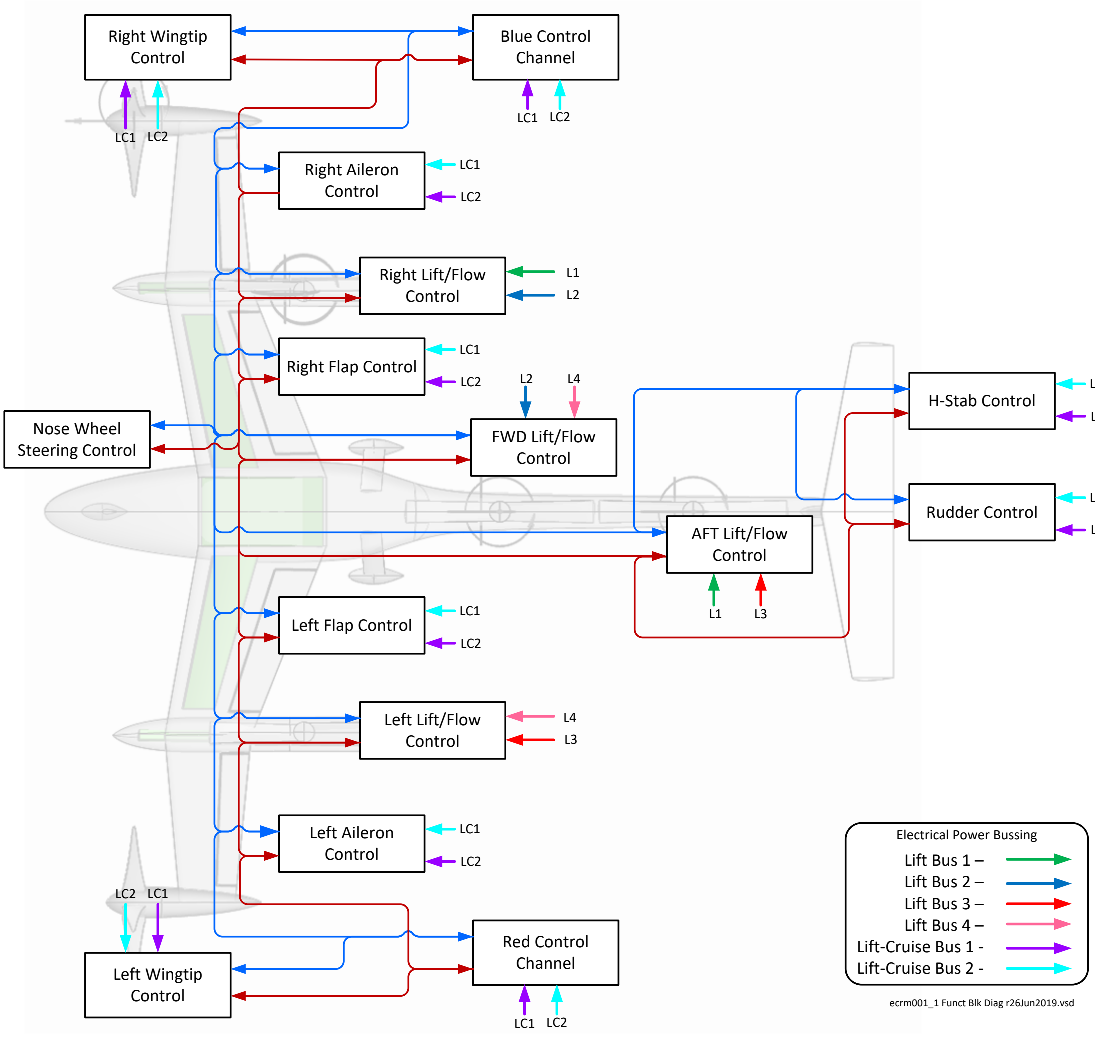


Figure A-11 eCRM-001 Flight-Propulsion Control Functional Block Diagram

Central Controller Group

The Central Controller Group is presented in Figure A-11 as the Blue and Red Control Channels located at the top center and bottom center of the figure. These two control channels provide the integrated and coordinated control capability needed to manage the various effector control groups for vertical and horizontal flight.

Wing Tip Control Group

A rotatable pod is installed at the end of each wingtip. Digitally controlled, electric motors rotate variable pitch propellers/rotors. When rotated to the vertical position, the wingtip pod motor-propeller combinations directly provide vehicle lift. When transitioned to the horizontal position, the wingtip pods provide forward thrust. The propeller pitch and pod positioning capabilities are also digitally controlled by the Central Controller Group functions.

Each Wingtip Control Function (detailed in Figure A-12 Wingtip Control Functional Block Diagram) provides fail-operational, fail-safe characteristics for most failures. Flight and propulsion control is provided by a single variable pitch propeller/rotor driven by dual-redundant electric motors summed through a mechanical gearbox. The electric motors are controlled by independent motor controllers, each interfaced to the Central Controller Group. Propeller/rotor blade pitch and nacelle tilt angle are each controlled by single channel electromechnical actuator (EMA) and controller combination.

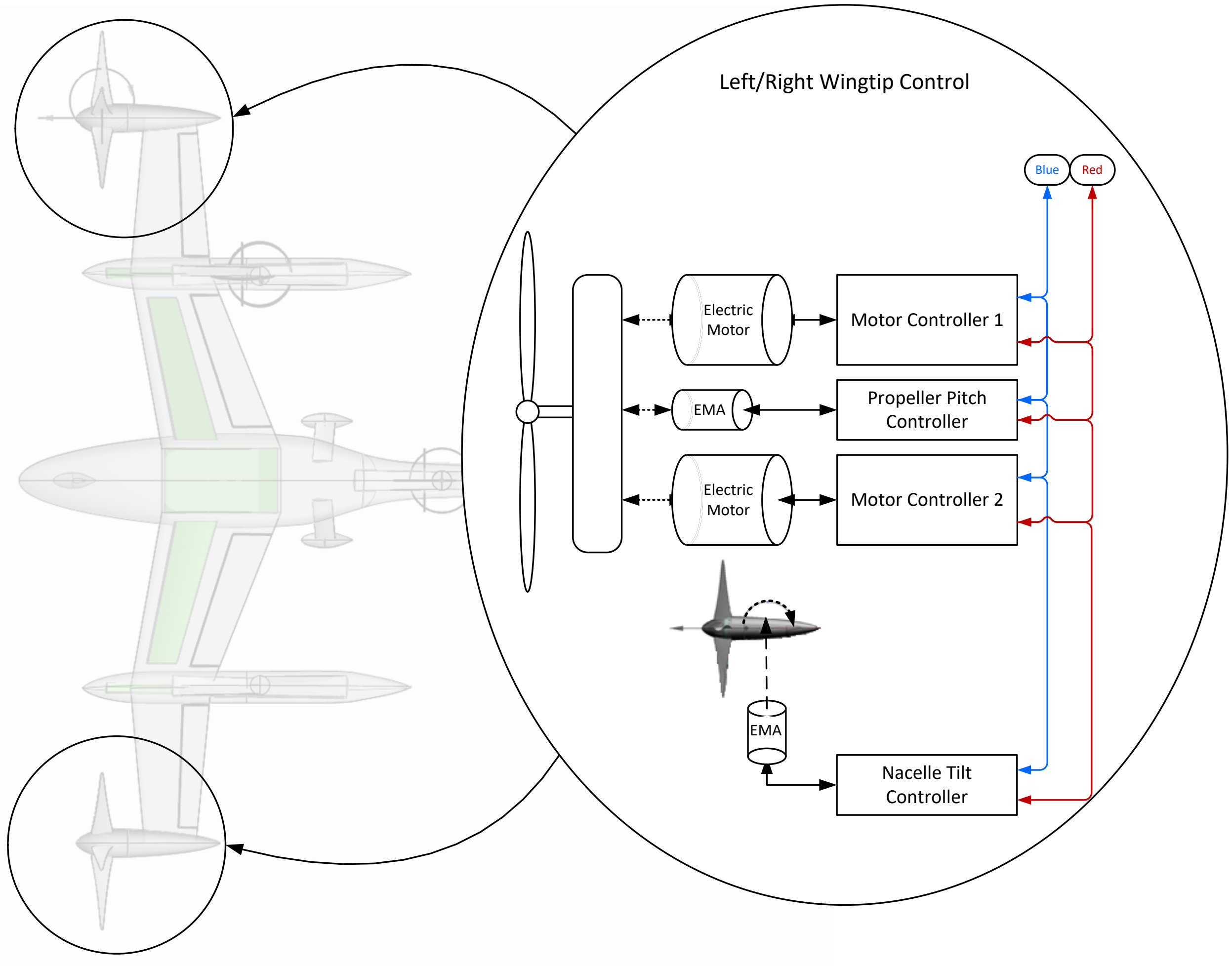


Figure A-12 Wingtip Control Functional Block Diagram

Lift Control Group

Right and left wing mounted, as well as fuselage mounted forward and aft electric motor driven rotors provide additional lift capacity during vertical flight phases. These lifting motor assemblies are stowable in wing pods or the fuselage, respectively, for horizontal flight. The right and left lift pods and fuselage lift rotors also contain rotor downwash flow control for additional yaw axis flight control management functionality.

The right/left wing and forward/aft Fuselage Lift Control Functions are presented in Figure A-13 Lift Control Functional Block Diagram. Dual co-rotating rotors are driven by dual-redundant electric motors mounted on a single shaft. The electric motors are controlled by independent motor controllers, each interfaced to the Central Controller Group. The rotor stow and flow control actuations are each controlled by single channel electromechnical actuator and controller.

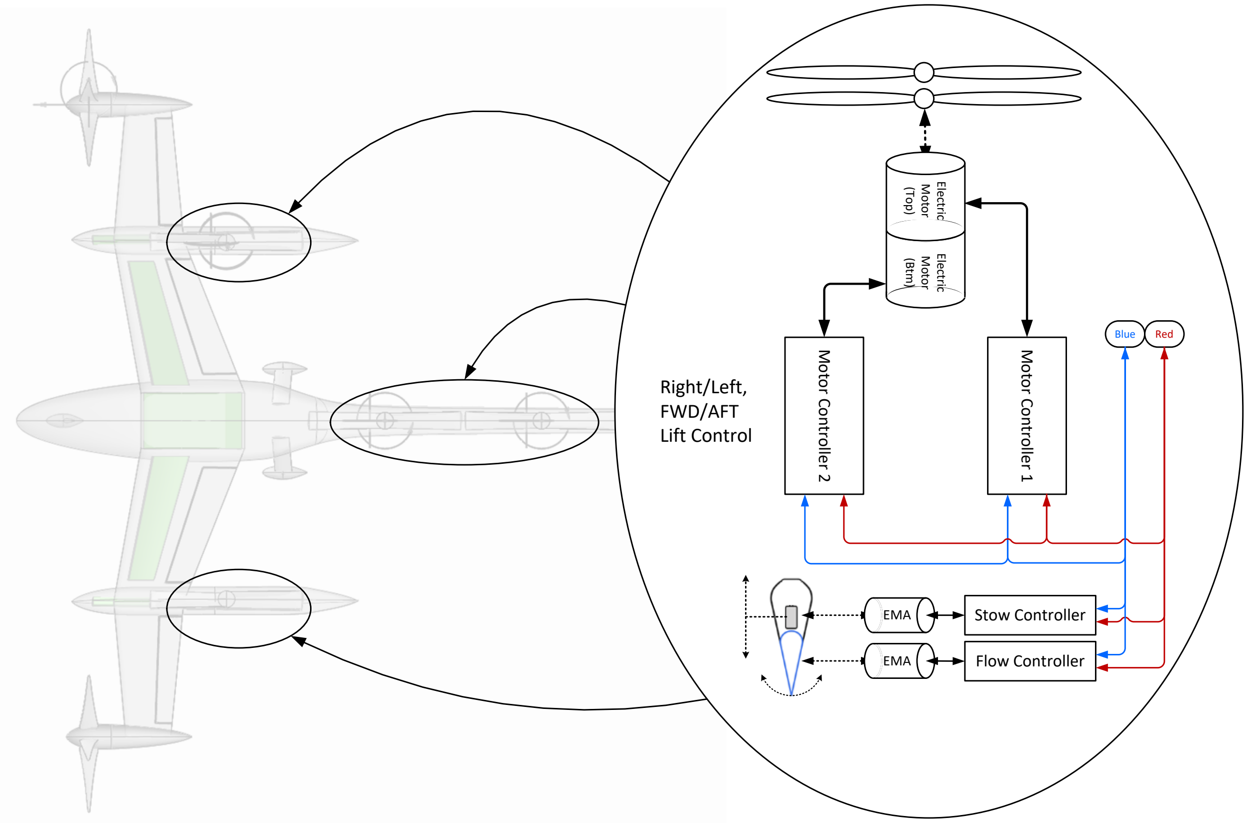


Figure A-13 Lift Control Functional Block Diagram

Surface Control Group

Conventional airplane aileron, flap, rudder and horizontal stabilizer control surfaces are provided for maneuvering during horizontal flight.

Conventional airplane surface control functional detail is presented in Figure A-14 Surface Control Functional Block Diagram. Each aileron, flap, rudder and the horizontal stabilizer is motivated by a single channel electromechanical actuator and controller combination interfaced with the Central Controller Group.

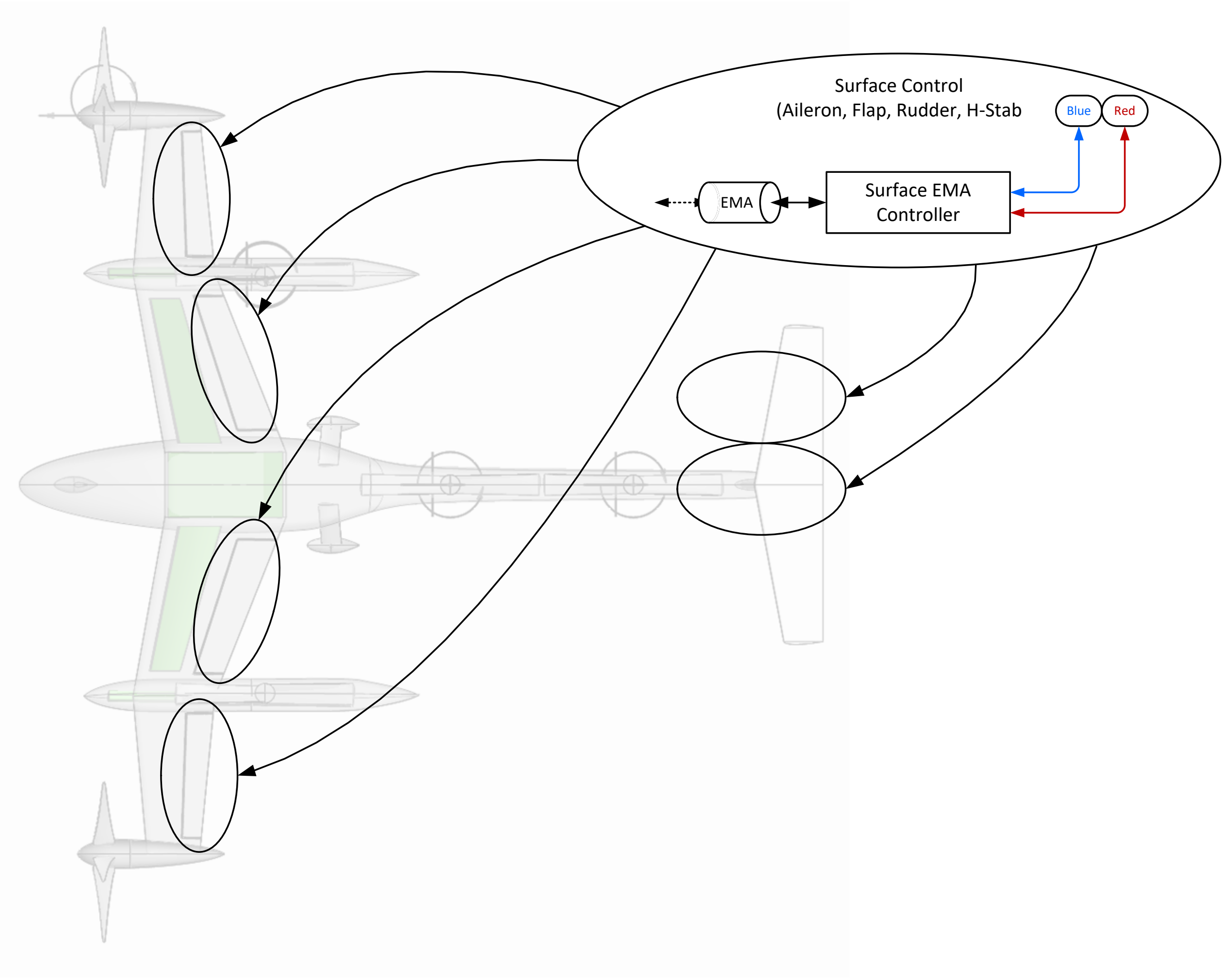


Figure A-14 Surface Control Functional Block Diagram

A2.6.2 eCRM-001 Electrical Power System

The eCRM-001 electrical power system is provided in both high-voltage and low-voltage formats. All electrical power will be sourced from onboard batteries which are architected in a manner to provide multiple independent sources. The independent high voltage and low voltage bus sources are then distributed to the FPCS loads.

Editor’s Note: While electric power is very important to an all-electric vehicle it has no further influence on the study outcomes so is omitted from other consideration for brevity.

A2.6.3 FPCS Architecture Safety Validation

A Multi-Function/Multi-System (MFMS) safety analysis is accomplished to validate the planned aircraft system architecture and develop any additional safety requirements. An extract from this MFMS analysis associated with the example focus function “Control VTOL Flight”, failure condition 2.1.1.TL “Loss of Vertical Lift MAC” is presented in Figure A-15.

The Figure A-15 extract presents the probability, per 0.5 hour flight, of loss of minimum acceptable lift during the landing phase. The analysis result confirms that the overall VTOL architecture strategy will satisfy the planned certification criteria (catastrophic for AW II ≤ 10^{-7} PFH) for the selected example function.

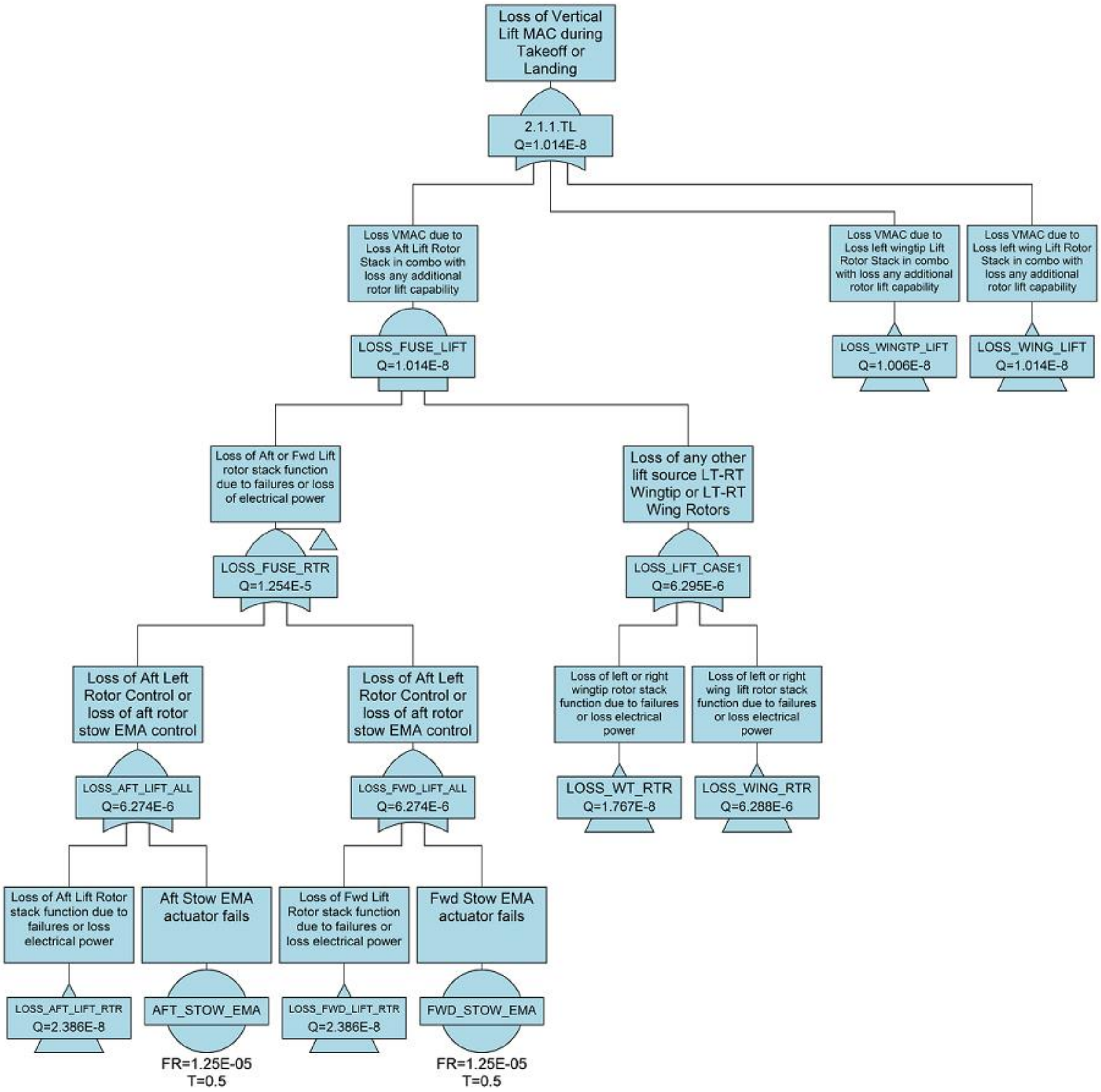

Loss of Vertical Lift MAC during Takeoff or Landing
2.1.1.TL
Q=1.014E-8
Loss VMAC due to Loss Aft Lift Rotor Stack in combo with loss any additional rotor lift capability
LOSS_FUSE_LIFT
Q=1.014E-8
Loss VMAC due to Loss left wingtip Lift Rotor Stack in combo with loss any additional rotor lift capability
LOSS_WINGTP_LIFT
Q=1.006E-8
Loss VMAC due to Loss left wing Lift Rotor Stack in combo with loss any additional rotor lift capability
LOSS_WING_LIFT
Q=1.014E-8
Loss of Aft or Fwd Lift rotor stack function due to failures or loss of electrical power
LOSS_FUSE_RTR
Q=1.254E-5
Loss of any other lift source LT-RT Wingtip or LT-RT Wing Rotors
LOSS_LIFT_CASE1
Q=6.295E-6
Loss of Aft Left Rotor Control or loss of aft rotor stow EMA control
LOSS_AFT_LIFT_ALL
Q=6.274E-6
Loss of Aft Left Rotor Control or loss of aft rotor stow EMA control
LOSS_FWD_LIFT_ALL
Q=6.274E-6
Loss of left or right wingtip rotor stack function due to failures or loss electrical power
LOSS_WT_RTR
Q=1.767E-8
Loss of left or right wing lift rotor stack function due to failures or loss electrical power
LOSS_WING_RTR
Q=6.288E-6
Loss of Aft Lift Rotor stack function due to failures or loss electrical power
LOSS_AFT_LIFT_RTR
Q=2.386E-8
Aft Stow EMA actuator fails
AFT_STOW_EMA
FR=1.25E-05
T=0.5
Loss of Fwd Lift Rotor stack function due to failures or loss electrical power
LOSS_FWD_LIFT_RTR
Q=2.386E-8
Fwd Stow EMA actuator fails
FWD_STOW_EMA
FR=1.25E-05
T=0.5

Figure A-15 MF/MS Analysis Extract – 2.1.1.TL

A2.7 – Development Process Objectives and Assumptions

In the revised 14CFR Part 23 [3] regulations, safety criteria for systems and equipment has been captured in §23.2510. Figure B-2 presents the 14CFR Part 23.2510 safety objective decoding process described in the referenced ASTM standards. This process identifies both the quantitative and qualitative safety objectives which must be satisfied for certification compliance.

The ASTM F3061 [16] guidance framework advises the applicant to identify an Aircraft Type Code (ATC) in order to select the appropriate development activities. The ATC is encoded as follows:

AW	N	T	S	C	M	A	F
Airworthiness Level	Number Of Engines	Engine Type	Stall Speed	Cruise Speed	Meteorological Conditions	Altitude	Flight Maneuvers

AW: In ASTM F3061 Table 1 [16], Airworthiness Level of the ATC is based on the number of passengers (pax) and crew the planned vehicle will transport ("1" -1 pax; "2" - 2-6 pax; "3" - 7-9pax; "4" – more than 10 pax). For the eCRM-001 study vehicle, the Airworthiness Level will be "2" due to the number of planned passengers (2-6 pax).

N: The number of engines employed on the aircraft is the next value encoded; "S" for single engine; "M" for multi-engine. The eCRM-001 will be "M" due to the use of multiple electric motors.

T: The engine type is assigned "R" for reciprocating or "T" for turbine per F3061. The eCRM-001 will be electrically powered which has yet to be considered in the F3061 guidance material. For this study, an engine type of "E" will be assigned.

The stall speed will be assigned based on the following criteria from F3061;

- L for a stall speed less than or equal to 83 km/h (43 knots),
- M for a stall speed greater than 83 km/h but less than or equal to 113 km/h (61 knots),
- H for a stall speed greater than 113 km/h (61 knots).

S: The eCRM-001 study vehicle has a stall speed of 80 knots resulting in stall speed assignment of "H".

C: The next character of the ATC is the cruise speed. For the eCRM-001 this will be assigned as "L" due to cruise speed of less than or equal to 463 km/h (250 knots).

M: The sixth character of the ATC is the planned meteorological operating conditions. This will be either "D" for Day VFR only; "N" for Day/Night VFR or "I" for instrument. The eCRM-001 is planned for Day VFR only so assigned "D".

A: The altitude character for the eCRM-001 will be assigned as "L", indicating a maximum operating altitude ceiling of equal to or less than 25000 feet.

F: The final character is assigned "N" for non-acrobatic flight maneuvers.

The resulting encoded eCRM-001 ATC would therefore be: 2 M E H L D L N. The appropriate sections of ASTM F3061 [16] may now be selected for assurance activity identification.

Editor's Note: Within the F3061 standard, the ATC is used to select many different aspects of vehicle development. This case study will focus only on the applicability, assignment and accomplishment of safety and development assurance activities specified in F3061 [16] section 4.2, System Safety Requirements.

Safety and development assurance activities of ASTM F3016 section 4.2 [16] are to be accomplished if the ATC indicates that the vehicle is Airworthiness Level 1, has a stall speed less than or equal to 45 knots (83 km/h) ("L") and is for Day VFR operating conditions ("D"). The eCRM-001 ATC indicates Airworthiness Level 2 indicating that all F3061 section 4.2 activities must be accomplished.

ASTM F3061 Section 4.2.3 identifies that implementation assurance activities must be accomplished to appropriate levels and these levels may be assigned per F3061 Table 2 or via the assignment methodology outlined in SAE ARP4754A [22] (However the ARP4754A approach would generally result in a higher hardware and software level assignments than those prescribed in ASTM F3061.

In order to identify the development assurance level objectives for airborne software (SW) and airborne electronic hardware (AEH), a Safety Assessment Level for the airplane must be determined per ASTM F3230 [17]. The safety assessment level in F3230 Table 3 uses the airworthiness level from the vehicle ATC, modulated by the type of engine (reciprocating or turbine) and quantity of engines. ASTM F3230 Table 3 is consistent in assigning a safety assessment level of "II" for more than one reciprocating engine or any quantity of turbine engines, indicating complexity may be a safety assignment factor. The eCRM-001 will utilize multiple electric motor propulsion technology "engines". Qualitatively the motors and their controllers will have the same complexity as the turbine engine case, so an Assessment Level of "II*"* is assigned.

Note: ASTM F3230 revisions are anticipated due to the development of ASTM F3338-18, "Standard Specification for Design of Electric Propulsion Units for General Aviation Aircraft" [21].

Appling the Assessment Level II assignment in ASTM F3061 Table 2, results in an airborne software (SW) and airborne electronic hardware (AEH) development assurance level objectives of level "C" for both primary and secondary systems.

The ASTM guidance documents do not prescribe a development assurance level associated with the airplane and system levels. However, ASTM F3061 section 4.1.5 [16] implies a need to capture or define intended system function. The capture of intended function enables verification of an implementation to be accomplished and establishes that the final implementation provides the defined function when installed and operated in defined installation environmental conditions.

The case study will therefore extend the AEH and software level assignments of "C" to the airplane and system level. At these levels, the objectives of an FDAL Level C ARP4754A [22] process will be accomplished to capture required airplane and flight-propulsion control function definitions.

Table A-3 summarizes the planned development assurance objectives by development level.

Table A-3 Development Assurance Assignment Summary

Hierarchy Level	Assignment Level - Standard
Airplane	FDAL C – ARP4754A objectives
System	FDAL C – ARP4754A objectives
Sub-System	FDAL C – ARP4754A objectives
Airborne Electronic Hardware	IDAL C – DO-254 objectives
Airborne Software	IDAL C – DO-178C objectives

In addition, ASTM F3230 section 4.2.4.3 prescribes that “no catastrophic failure condition should result from failure of a single component, part or element of a system.” The allowable quantitative probability for combinations must be shown to meet $<10^{-7}$ per flight hour or per flight phase.

Editor’s Note: EASA SC-VTOL-01 [11] mandates an alternative assignment for function, system and item level development assurance. FDAL and IDAL assignments of Level “A” are required for the Enhanced Category (Distinguished from Basic Category by operating over congested areas or for commercial operations). The potential consequences of this alternate assignment are discussed in section A6.0.

A3.0 System Development and Planning

An extract from the system development plan is provided in Figure A-16 Integrated Flight Propulsion Control System Requirement Capture Strategy. Requirements for flight and propulsion control functions will be captured at the airplane level in an FPCS Specification document. The integrated flight and propulsion control function control laws and logic will be independently captured in another document (eCRM-001 FPCS Control Laws).

Requirements for the FCC and each of the distributed controllers will be discretely captured and progress to greater levels of detail in the applicable DO-254 [26] and DO-178C [25] process artifacts (color coded gold and blue, respectively).

The functional development assurance level objectives of ARP4754A will be accomplished for the FPCS and sub-system development levels (i.e. FCC, motor control system requirement specifications (SRSs) color coded as yellow and green)

Editor’s Note: A project decision to reduce the number of requirement capture documents was made such that some control functions were combined (e.g. aileron and flap control into Aileron/Flap Control SRS).

Editor’s Note: The application of mode based software development, when contemplated, would add satisfaction of DO-331 [27] processes and artifacts.

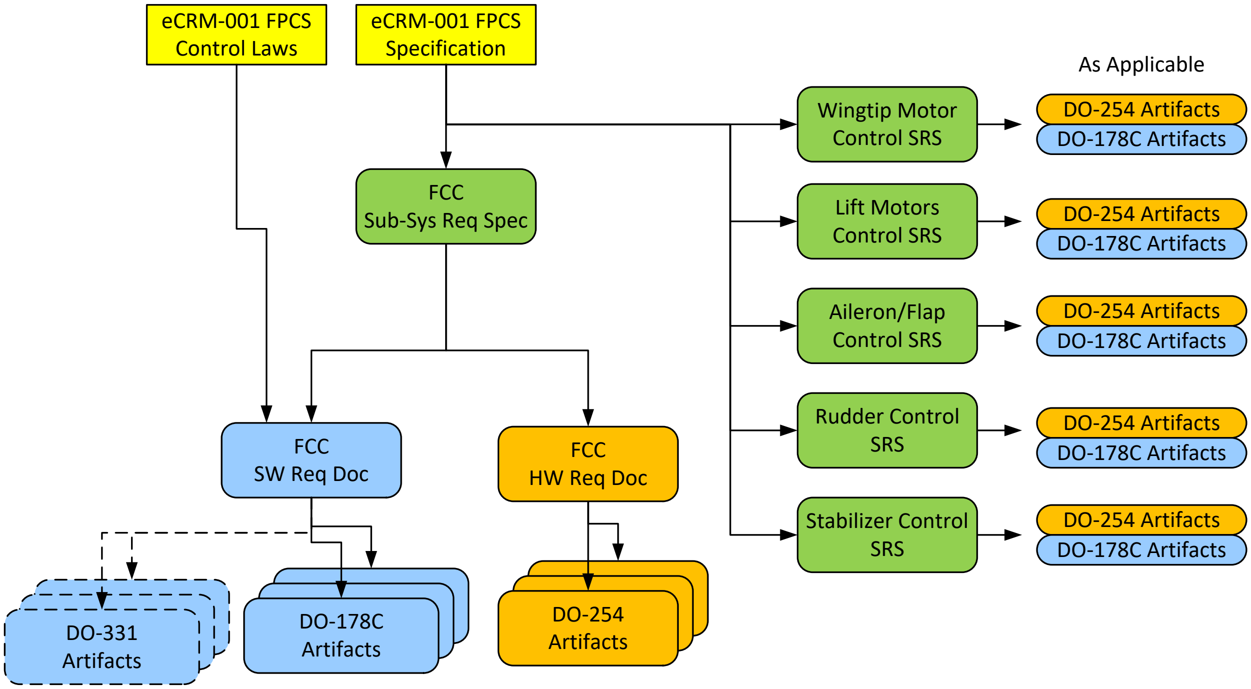


Figure A-16 Integrated Flight Propulsion Control System Requirement Capture Strategy

A3.1 Develop FPCS Level Functions

The bulk of the FPCS high level function control, shown in Figure A-17, will be allocated to the Central Controller Group and the Flight Control Computers (FCCs). Each FCC will provide the following functions:

- Sensor & Sensor Management – Provide pilot controls and aircraft sensor interface data management and provide sensor validity monitoring,
- Effector & Effector Management – Provide effector controls and effector validity monitoring,
- FPCS State Management – Provide management of FPCS and FCC state based on the validity monitoring of sensors, effectors and internal computer status,
- FPCS Control Laws – Provide closed loop movement control of vehicle.

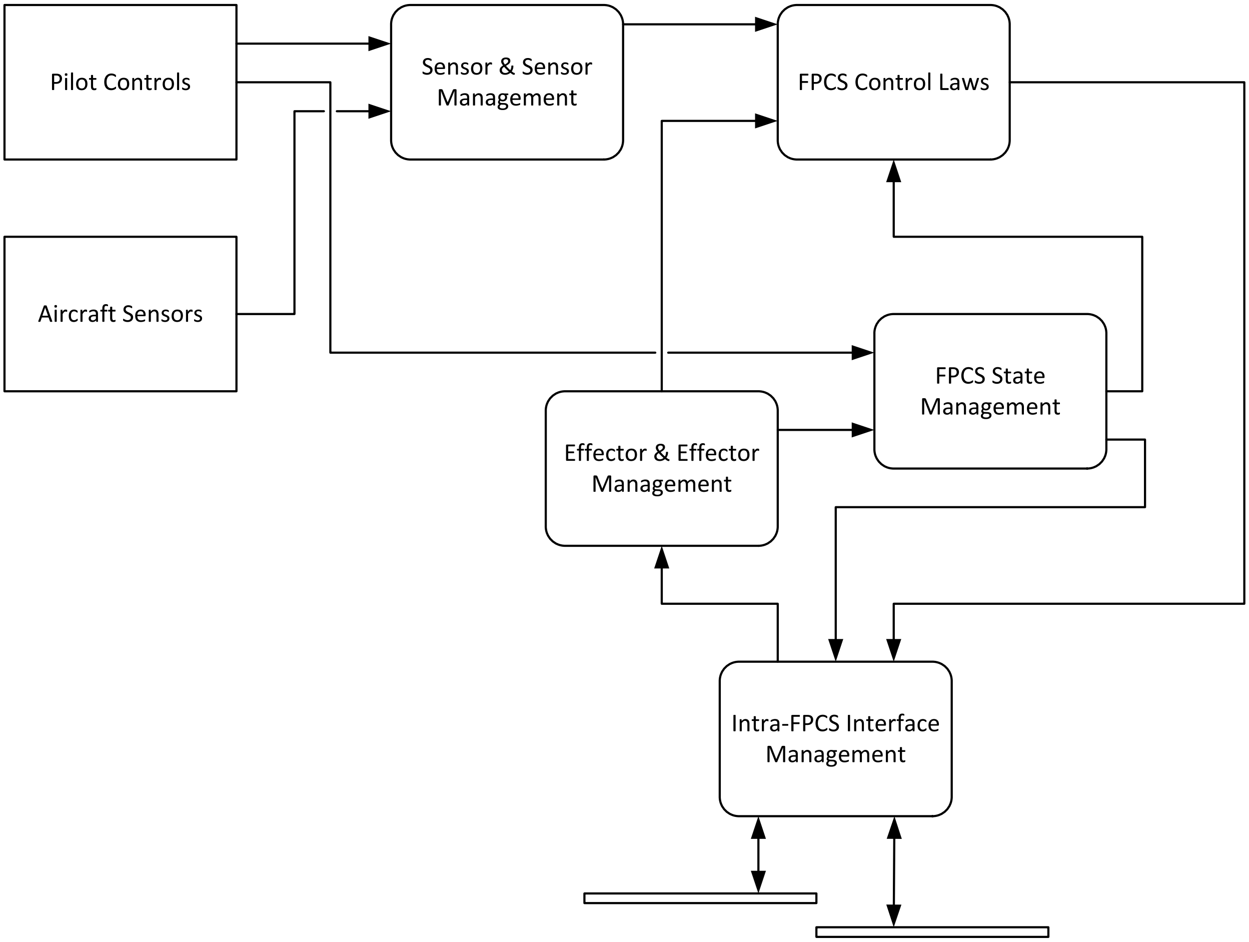

Pilot Controls
Sensor & Sensor Management
FPCS Control Laws
Aircraft Sensors
FPCS State Management
Effector & Effector Management
Intra-FPCS Interface Management

Figure A-17 HL System Function Diagram

Editor's Note: The FPCS Control Law and FPCS State Management functions will be the continued focus of the example description.

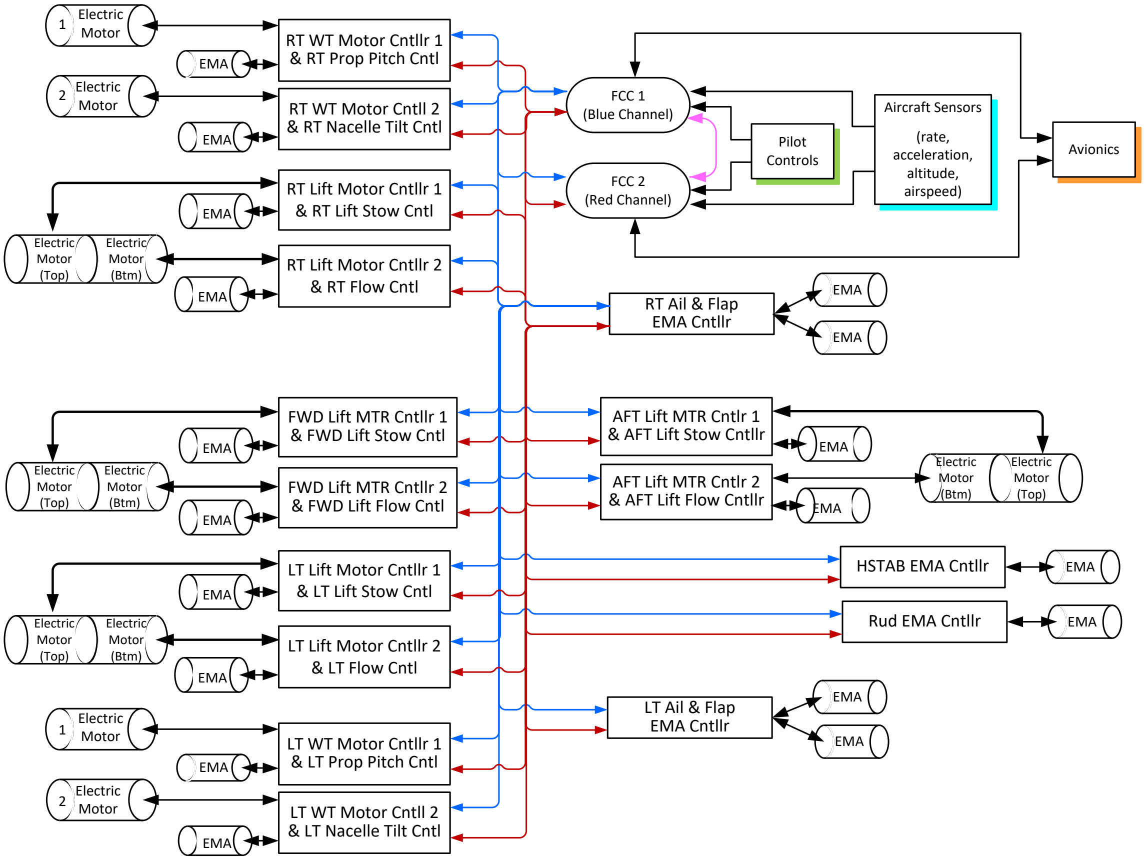
Table A-4 Flight Propulsion System Function Allocations – Initial presents the initial system function allocations based on the preliminary architecture established in Figure A-11 eCRM-001 Flight-Propulsion Control Functional Block Diagram. The initial allocation is further refined and combined to create a more efficient and weight conscious solution. Table A-5 Flight Propulsion System Function Allocations – Final Allocations presents the updated functional allocations.

Figure A-18 Revised eCRM-001 Flight Propulsion Control System Block Diagram presents the consolidated Flight-Propulsion Control System architecture based on the final functional allocations.

Editor's Note: The flow control effectors, which provide additional translation control during vertical flight, have been omitted from further discussion for example brevity.

Table A-4 Flight Propulsion System Function Allocations - Initial

1	2	3	4	5	6	7	8	9	10	11
System Element	**Wingtip Control Group**					**Lift Control Group (LT, RT, FWD, AFT)**				**Central Control Group**
Function	**Motor Control 1**	**Motor Control 2**	**Rotor Pitch Control**	**Nacelle Tilt Control**	**Gear Box**	**Motor Control 1**	**Motor Control 2**	**Stow Control**	**Common Shaft**	**FCC**
Control Lift:										
Control left/right wingtip rotor lift	X	X	X	X	X					X
Control left/right wing rotor lift						X	X	X	X	X
Control forward/aft fuselage rotor lift						X	X	X	X	X



1 Electric Motor
EMA
2 Electric Motor
EMA
RT WT Motor Cntllr 1 & RT Prop Pitch Cntl
RT WT Motor Cntll 2 & RT Nacelle Tilt Cntl
EMA
Electric Motor (Top)
Electric Motor (Btm)
EMA
RT Lift Motor Cntllr 1 & RT Lift Stow Cntl
RT Lift Motor Cntllr 2 & RT Flow Cntl
FCC 1 (Blue Channel)
FCC 2 (Red Channel)
Pilot Controls
Aircraft Sensors (rate, acceleration, altitude, airspeed)
Avionics
RT Ail & Flap EMA Cntllr
EMA
EMA
FWD Lift MTR Cntllr 1 & FWD Lift Stow Cntl
FWD Lift MTR Cntllr 2 & FWD Lift Flow Cntl
AFT Lift MTR Cntlr 1 & AFT Lift Stow Cntllr
AFT Lift MTR Cntlr 2 & AFT Lift Flow Cntllr
Electric Motor (Btm)
Electric Motor (Top)
HSTAB EMA Cntllr
Rud EMA Cntllr
LT Lift Motor Cntllr 1 & LT Lift Stow Cntl
LT Lift Motor Cntllr 2 & LT Flow Cntl
LT Ail & Flap EMA Cntllr
LT WT Motor Cntllr 1 & LT Prop Pitch Cntl
LT WT Motor Cntll 2 & LT Nacelle Tilt Cntl

Figure A-18 Revised eCRM-001 Flight Propulsion Control System Block Diagram

Table A-5 Flight Propulsion System Function Allocations - Final

1	2	3	4	5	6	7	8	9	10	11
System Element	**Wingtip Control Group**					**Lift Control Group (LT, RT, FWD, AFT)**				**Central Control Group**
Function	**Motor 1 & Rotor Pitch Control**	**Motor 2 & Nacelle Tilt Control**			**Gear Box**	**Motor 1 & Stow Control**	**Motor 2 & Flow Control**		**Common Shaft**	**FCC**
Control Lift:										
Control left/right wingtip rotor lift	X	X			X					X
Control left/right wing rotor lift						X	X		X	X
Control forward/aft fuselage rotor lift						X	X		X	X

A3.2 Develop System Level Architecture Requirements

The flight and propulsion control system is distributed across the airframe. This architecture features dual redundant distributed control for all vehicle propulsion and vertical flight control aspects, with single path control of the nominal “airplane mode” aileron, rudder, flap, and horizontal stabilizer effectors. Coordinated control of distributed controllers is managed by a pair of Flight Control Computers (FCCs).

The FCCs communicate via high speed bi-directional digital data busses with the distributed controllers to manage the various flight and propulsion control assets. The FCCs receive pilot control inputs and interface to aircraft sensors for closed loop airplane stability management.

The two FCCs operate in an “active-standby” arrangement whereby only one FCC is actively managing control of vehicle flight and propulsion control effectors at any time. Upon detected failures of an active FCC, the active FCC relinquishes control to the standby FCC, which assumes the active control state. The FCCs will alternate operating in active or standby control modes, nominally decided at vehicle power-up on the ground to reduce latent failure detection latencies.

Editor’s Note: An alternative active/standby time managing interval may be determined by the preliminary system safety assessment (PSSA).

Each FCC interfaces with an independent and redundant set of aircraft state sensors. Figure A-19 FCC Sensor Interface Block Diagram presents a high level sensor interface diagram for the FCCs. Each FCC interfaces directly with air data sensors, attitude sensors, heading sensors, rate sensors and AOA sensors. It is anticipated that vehicle control will require the following vehicle information:

- Airspeed,
- Altitude,
- Attitude (pitch, roll, yaw)
- Rate (pitch, roll, yaw),
- Angle of Attack (AOA).

Each FCC also receives a duplicate set of "off-side" sensor interface information via an FCC to FCC digital data link. This information will be used for failure isolation algorithms.

Editor's Note: Conventional sensor, motor, and computer management techniques would be applied to manage physical implementation failures in both baseline as well as the alternate RTA approach.

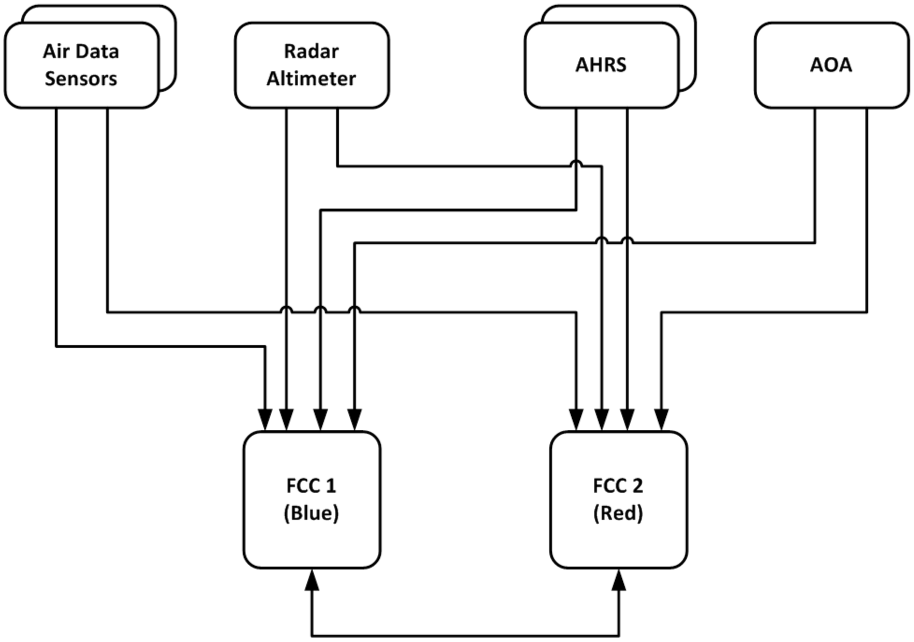


Figure A-19 FCC Sensor Interface Block Diagram

A3.2.1 Develop Example Manual VTOL Control Function Requirements

Due to the large number of rotating elements, control surfaces, and other actuators, it is an imperative that some level of automation and stabilization exists for manual control. A reasonable control approach is to have the pilot use an inceptor to command desired roll and pitch angles (called Attitude Command Attitude Hold or ACAH), with feedback control and an allocator designed to achieve those commands. Additional inceptors, such as a lever and pedals, could be used to command vertical velocity and yaw rate, respectively. With training, pilots would be capable of accurate aviation using this approach. This control approach would be designed and implemented using a traditional assurance approach.

However, studies have suggested that translation-rate command (TRC), in which the pilot uses an inceptor to command a horizontal velocity vector rather than an attitude, may be a better approach for pilots with lesser training. In this design, vertical velocity and yaw rate would be commanded in the same manner as with ACAH. This control architecture adds a layer of complexity on top of a basic ACAH controller, and it would constitute a higher level of automation. The ATCorp Team is not aware of any commercial aircraft that have been certified and that employ this control approach. Figure A-20 presents this combined control approach.

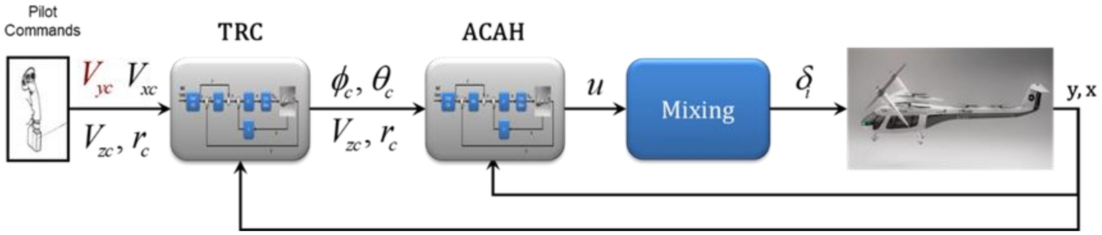


Figure A-20 Manual Control Approach

This basic pattern can be extended to higher levels of automation as shown in Figure A-21. Hovering flight mode, employed during the initial and final flight segments, requires a high level of energy expenditure because the vast majority of lift is produced directly from rotating surfaces of the aircraft. It is desirable to minimize the amount of time spent in Hover Mode to conserve electrical energy and maximize the number of passenger carrying flights before a full recharge cycle is needed. An autoland control algorithm will be developed specifically for this purpose.

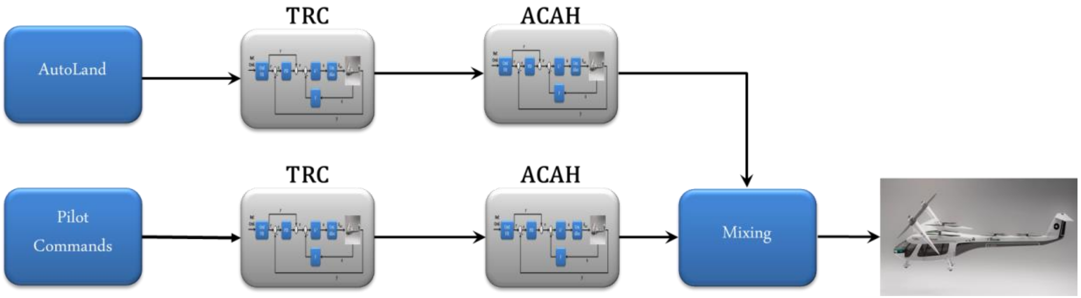


Figure A-21 Baseline Manual and Automated Control Approach

The automatic control approach and specifically autoland scenario will be the detailed focus of further example development.

A3.2.2 Develop Example Automatic VTOL Control Function Requirements

This section develops the requirements of an autoland control algorithm, assuming the traditional certification approach. The autoland algorithmic structure is shown in Figure A-22. In operation, the onboard pilot will instruct the autoland component to set the aircraft down at a specified location. The onboard pilot then monitors the progress of the maneuver and should be able to resume direct control of the vehicle at his / her discretion.

Reference commands from the autoland control algorithm specify lateral velocity (V_{yc}), longitudinal velocity (V_{xc}), vertical velocity (V_{zc}), and yaw rate (r_c) to the TRC module. The TRC passes the vertical velocity and yaw rate command directly to the ACAH controller and determines the required roll (ϕ_c) and pitch (θ_c) attitude commands to track the velocity command. The ACAH controller determines the control signal, u, that follows the commanded attitudes, vertical velocity, and yaw rate. A mixing, or control allocation, system then maps the command signal to individual surfaces depending on the vehicle configuration and flight mode. Aircraft states, x, and measurements, y, are provided to the controllers.

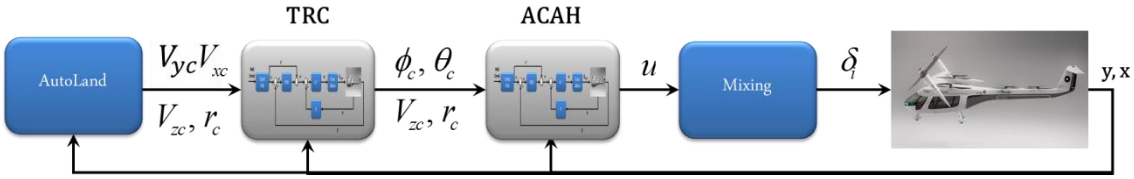


Figure A-22 Autoland Algorithmic Structure

Both the TRC and ACAH algorithms are based on a dynamic inversion design. Figure A-23 shows a general architecture for this kind of design. Dynamic inversion (DI) controllers generate a command; u, that follows a reference command generated by a command filter. The nonlinear dynamic inversion (NLDI) control approach includes nonlinear kinematics in the inversion approach and can streamline the control design process by reducing the need for gain scheduling. The NLDI approach is based on the concept of feedback linearization [106] wherein the actual plant dynamics are inverted, or approximately so, to command the plant to follow a set of desired dynamics. DI has been successfully applied to fixed-wing vehicles [107][108], rotorcraft [109], and tiltrotors [110]. Consider a system described by Eq. 1.

$$\begin{gathered} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{gathered} \qquad \text{Eq 1}$$

where $\boldsymbol{x} \in \Re^n$ are the states; $\boldsymbol{u} \in \Re^m$ is the control vector; and $\boldsymbol{y} \in \Re^m$ is a vector of measured outputs called the controlled variables (CV). The number of outputs equals the number of control elements in $\boldsymbol{u}$ and full-state feedback is required in addition to the measurement vector. The reference command passes through a command filter, sometimes called a flying qualities model, which generates the reference command used in the design, ***r*,** and its derivative.

The command filter generates a desired response to pilot inputs and is designed following handling quality standards like ADS-33 [111] for the vertical flight phases and MIL-STD-1797 [112] for forward flight phase operation. The DI controller follows the reference command by calculating the control signal u (Eq. 2):

$$u = G^{-1}(x)(\nu - F(x)) \qquad \text{Eq 2}$$

So the closed-loop dynamics are expressed by $\dot{x} = \nu$, where ν is called a pseudo-control. Here, $F(\boldsymbol{x})$and $G(\boldsymbol{x})$ are approximations to the actual dynamics. If the vehicle dynamics were perfectly modeled and there were no disturbances, the vehicle would follow the desired response: ν. In practice, these assumptions do not exist and a linear PID compensator, K, is added to ν to track the desired signal and govern disturbance compensation.

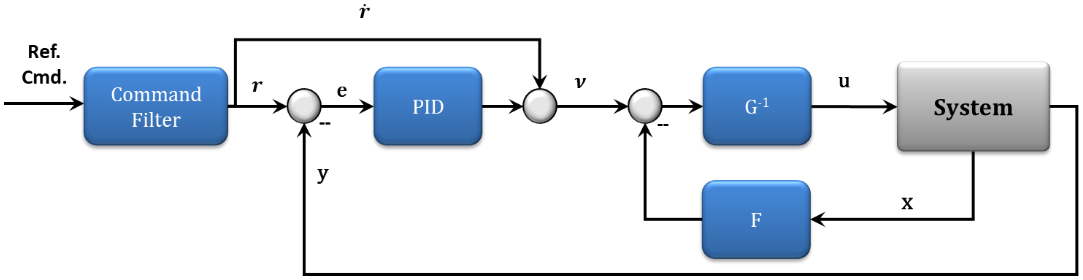


Figure A-23 Dynamic Inversion Architecture

The DI scheme requires the output vector, **y**, to be differentiated until the control signal appears in the resulting equation and that $G(\boldsymbol{x})$ is invertible. Differentiating **y** results in Eq. 3.

$$\dot{y} = \frac{\partial h}{\partial x}(x)\dot{x} = F(x) + G(x)u$$
$$F(x) \triangleq \frac{\partial h}{\partial x}(x)f(x), \quad G(x) \triangleq \frac{\partial h}{\partial x}(x)g(x) \qquad \text{Eq 3}$$

where $F(\boldsymbol{x})$ and $G(\boldsymbol{x})$ are the Jacobians of the nonlinear system evaluated at different flight conditions. Substituting Eq. 2 into Eq. 3 results in a system of decoupled integrators: $\dot{y} = \nu$. The pseudo-control, ν, is defined by the reference signal and a linear compensator acting on the error signal (Eq. 4):

$$\nu = \dot{r} + K(s)e$$
$$e \triangleq r - y \qquad \text{Eq 4}$$

The error dynamics determine the response to disturbances and modeling errors and are described as Eq. 5:

$$\dot{e} = \dot{r} - \dot{y} \;=\; \dot{r} - v \qquad \text{Eq 5}$$

The error dynamics are therefore stable with a proper choice of ν and compensator K(s), which depends on the choice of controlled variables composing the output vector $\boldsymbol{y}$.

ACAH Requirements

The ACAH controller is based on the NLDI variant of DI control discussed above. Recall that the controlled variables for the ACAH controller are: $\boldsymbol{y} = \{\dot{\phi}, \dot{\theta}, V_z, r\}^T$. Note that, although the TRC algorithm commands an attitude, the controlled variables are attitude rates. The commanded attitudes can still be tracked by proper design of the command filter. Alternatively, the attitudes can be selected as the command variables but in this case the output vector must be differentiated a second time. The selection of attitude rates as the command variables results in a cleaner design [109]. The states and control vectors are presented in Eq. 6.

$$\begin{aligned} \boldsymbol{x} &= \{u, v, w, p, q, r, \phi, \theta, \psi\}^T \\ \boldsymbol{u} &= \{\delta_{lat}, \delta_{lon}, \delta_{col}, \delta_{ped}\}^T \end{aligned} \qquad \text{Eq 6}$$

The states are from the rigid-body equations of motion and include the body-frame velocities (u, v, w), angular rates (p, q, r), and Euler angles (ϕ, θ, ψ). In traditional rotorcraft applications, the control vector includes signals for lateral and longitudinal cyclic, collective, and pedal. For this UAM application, these correspond to control in each translational axis and heading. A control allocation, or mixing, system converts these commands to actuator commands. Eq.7 is the expression of the output vector given as a nonlinear function of the states.

$$y = h(x) = \begin{bmatrix} p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ q\cos\phi - r\sin\phi \\ u\sin\theta - v\sin\phi\cos\theta - w\cos\phi cos\theta \\ r \end{bmatrix} \qquad \text{Eq 7}$$

The control architecture for the inner-loop is shown in Figure A-24.

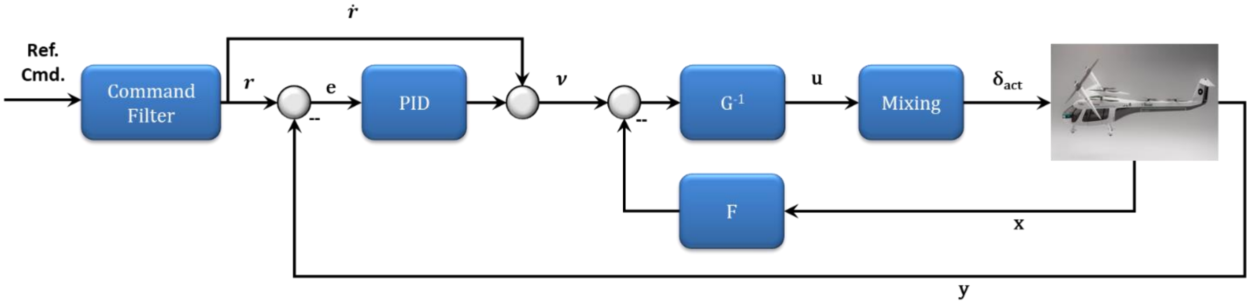


Figure A-24 ACAH Algorithmic Structure

For the roll and pitch channels, the Command Filter is a second-order system and the compensator K(s) is chosen as a proportional plus integrator plus double integrator system. The structure of the Command Filter and compensator for the pitch command is shown in Figure A-25. In the heave axis (i.e. up/down), a rate command / height hold (RCHH) DI controller is used and a rate command / direction hold (RCDH) DI controller is used in the yaw axis. These axes use a first-order command filter as shown in Figure A-26 for the heave axis and a PI compensator. The command filter parameters, ω_n and ζ, govern the response to command inputs and are chosen according to ADS-33 [111] to achieve Level 1 Handling Qualities in each axis.

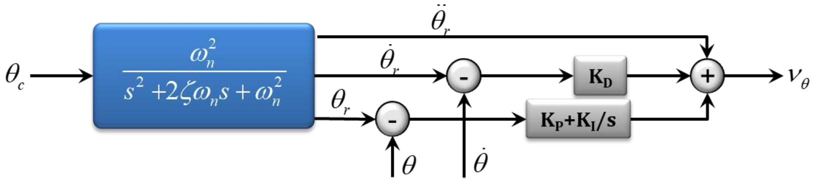


Figure A-25 Second-order Command Filter

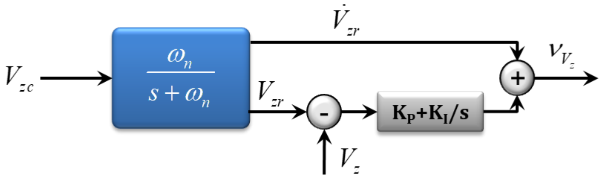


Figure A-26 First-order Command Filter

Recall the error dynamics in Eq. 5, which we can now rewrite for the pitch and roll attitude errors as Eq. 8:

$$\dot{e} = K(s)e = (K_p + \frac{K_I}{s} + \frac{K_{II}}{s^2})e$$
$$e(s)(s^3 + K_p s^2 + K_I s + K_{II}) = 0 \qquad \text{Eq 8}$$

We can factor the error dynamics in Eq. 8 to be in the form of a single real pole and second-order system as Eq. 9:

$$(s+p)(s^2+2\zeta\omega_n s+\omega_n^2) \qquad \text{Eq 9}$$

Equating the coefficients of the polynomials in Eq. 8 and Eq. 9 results in the gains presented in Eq. 10.

$$K_p = 2\zeta\omega_n + p, \quad K_I = 2\zeta\omega_n p + \omega_n^2, \quad K_{II} = \omega_n^2 p \qquad \text{Eq 10}$$

These are the gains for the compensator in each axis and govern the bandwidth response to disturbances in a similar manner as the command filters set the response bandwidth to commanded inputs. As with other control approaches, gain selection is a tradeoff between robust stability and performance.

TRC Requirements

For the TRC loop, a linear DI design is used, which could be augmented to include position-hold functionality. Its inputs are velocities commanded in the longitudinal and lateral axes in the *vehicle heading* frame, vertical rate in the heave axis, and heading rate in the yaw axis. With a neutral input (e.g., pilot control inceptors are at trim), the current 3D position and heading angle are maintained. The vehicle heading frame is the NEU frame rotated to align with the aircraft's current heading. The x-axis in the vehicle heading frame points in the current heading and the y-axis points out the right wing. In this case, the inertial North (N), East (E), and Up (U) position are controlled. A schematic of the controller for the N position is shown in Figure A-27.

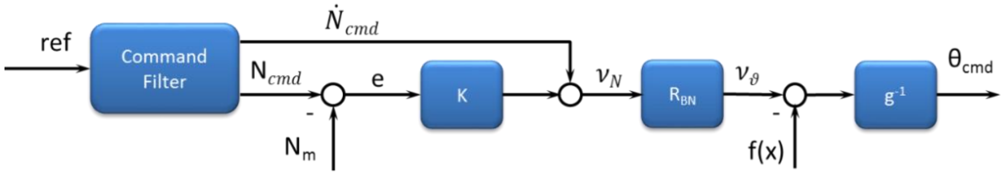


Figure A-27 TRC / Position Hold Algorithmic Structure

The vehicle dynamics describing translational motion are based on pitch and roll changes from trim (Eq. 11):

$$\begin{aligned} \dot{V}_x^{vh} &\approx -g\theta \\ \dot{V}_y^{vh} &\approx g\phi \end{aligned} \qquad \text{Eq 11}$$

Using this model, the pitch and roll attitude are used as commands to the inner-loop controller to regulate inertial position. The control signals are given by Eq. 12.

$$\theta_{cmd} = -\frac{1}{g}(K_{Px}\tilde{x} + K_{Ix}\int \tilde{x}\,\mathrm{dt} + \mathrm{K}_{Dx}\dot{\tilde{x}})$$
$$\phi_{cmd} = \frac{1}{g}(K_{Py}\tilde{y} + K_{Iy}\int \tilde{y}\,\mathrm{dt} + \mathrm{K}_{Dy}\dot{\tilde{y}}) \qquad \text{Eq 12}$$

Here $\tilde{x}$ and $\tilde{y}$ are the North and East errors rotated in the vehicle heading frame. These commands are input to the inner-loop controller which calculates actuator commands to track the commanded pitch and roll attitudes.

Autoland Requirements

The autoland system is a flight control mode that enables automated landing of the UAM vehicle in a congested urban setting. It consists of multiple components required to support the high-level execution of the landing function, including:

- Navigation algorithms for determining the location and orientation of the aircraft within the environment,
- Path planning algorithms that generate a safe and feasible trajectory to the landing point; and
- Guidance laws that generate translation rate commands to follow the chosen path.

The block is supported by a variety of both exteroceptive[2] and proprioceptive[3] sensors for detecting obstacles, measuring distances to external objects, quantifying turbulence and other atmospheric conditions, etc. Certification of these sensors and their controlling software is beyond the scope of this effort.

An abbreviated list of the functions implemented in the autoland system includes:

1. Determine a dynamically feasible and safe path to follow to the landing site.
2. Ensure the path is free from obstacles.
3. Confirm the vehicle maintains a required safe state from obstacles throughout the maneuver. Here, safe state is determined by a combination of distance and impact time or time to breach a safety volume.
4. Define a safety tube within which the vehicle should remain for the entire maneuver.
5. Ensure the exteroceptive sensors can see the landing site for the final approach portion.
6. Track objects in the sensor field-of-view that may interfere with the flight path.
7. Monitor vehicle state to ensure it does not enter danger zones like vortex ring state.

Note that each of these functions requires a suite of properly functioning sensors and systems that may not be certified for traditional piloted aviation or certified at a lower level than is necessary for UAM operations. In the following, a synopsis of the Path Planner is provided and with a slight abuse of terminology, path planning and trajectory generation are used interchangeably.

[2] sensing external aircraft characteristics.
[3] sensing internal aircraft characteristics especially those connected with position and movement.

Path Planner

The role of the path planner is to generate a dynamically feasible and safe path from the current vehicle position to the designated landing site. Systems estimating static and dynamic obstacles, health monitoring, and navigation systems are inputs to the path planner. A number of methods have been used to solve path-planning problems including graph-based approaches such as Dijkstra's Algorithm, A*, LRTA*, D*; Monte Carlo or sample-based techniques such as probabilistic roadmaps (PRM) [113] and Rapidly-Exploring Random Trees (RRT) [114]; potential fields, Voronoi diagrams, and optimal control formulation including dynamic programming and mixed integer linear programming (MILP) [115][116].

Optimal control formulations, which rely on techniques such as variational calculus or the minimum principle of Pontryagin, etc., become computationally expensive, and in some cases intractable, for an increased number of vehicle states and constraints. To solve deterministic and complete algorithms requires exponential time in the state-space dimension of the dynamic system and polynomial time in the number of obstacles [117]. Therefore, deterministic and complete algorithms are usually implemented for systems with low dynamic dimensions. Even in the single vehicle case the “curse of dimensionality" arises when vehicle dynamics and differential constraints are included (kinodynamic or nonholonomic motion planning) [114].

These methods can generally be categorized in terms of completeness, computational complexity, optimality, and scalability. Completeness refers to a method's ability to find a path assuming one exists or indicates there is not a feasible path. Computational complexity provides a relative time metric and can help determine if a method can be used online for a particular application. Optimality indicates the method finds a path that minimizes a specified cost or criterion. Scalability is the ability to implement the planning method in various systems, including complex and high-dimensional systems.

An illustrative example of a path planner that includes constraints on the boundary conditions, acceleration constraints throughout the maneuver, and can plan to avoid dynamic obstacles follows a sequential convex programming approach [118].

The path planning procedure follows the primary steps:

1) The initial and final position, velocity, and acceleration are defined over a fixed time horizon, T, as equality constraints.

2) Additional convex constraints are defined for the maximum and minimum velocity, acceleration, and jerk. Position constraints can be defined as minimum and maximum values in some coordinate frame or as a convex polytope describing an obstacle-free region. Non-convex constraints are defined for collision avoidance of other vehicles and obstacles.

3) The optimizer discretizes the transition path into K steps corresponding to a desired discrete time step, h. The optimizer will adjust the acceleration of the vehicle at every time step to satisfy constraints on position, velocity, acceleration, and jerk. The objective function is the norm of the acceleration over all K time steps resulting in a minimum-power path. The problem can also be formulated as a minimum-time problem or mixed-norm function.

The convex constraints can be reasonably integrated into a quadratic programming architecture. However, collision avoidance requirements of either other vehicles or obstacles are non-convex constraints and cannot be natively integrated into the quadratic optimization. The SCP approach is to linearize the non-convex constraints about a prior solution. The resulting affine functions can be incorporated as conventional linear constraints into the optimization. In practice, the initial vehicle trajectory solution is computed without considering the non-convex collision-avoidance constraints. The trajectory optimization is repeated with the convex constraints linearized about the initial trajectory result to incorporate linear collision avoidance constraints. Subsequent calls to the optimizer continue to use the most recent solution for linearizing the non-convex collision avoidance constraints. The optimization converges once a successful trajectory is obtained which satisfies the constraints and changes little on successive calls to the optimizer. Therefore, to implement collision avoidance constraints the overall trajectory optimization is repeated multiple times. However, the quadratic optimization with linear constraints converges extremely rapidly and the entire solution can be obtained in a fraction of a second.

Two examples of the path planner are shown in a generic environment. Figure A-28 presents the first scenario where the vehicle has an initial position at (50, 25, 80) and no velocity. A single obstacle is represented by the red sphere. The task is to plan a path at the origin while satisfying position, velocity, and acceleration constraints and avoiding the obstacle. Two perspectives are presented in the figure. The black line indicates the optimal, nearly straight-line path that was planned if there were no obstacles present. The blue line is the optimal path with the obstacle present which can be avoided by virtue of the vertical flight ability by simply descending beneath the obstacle.

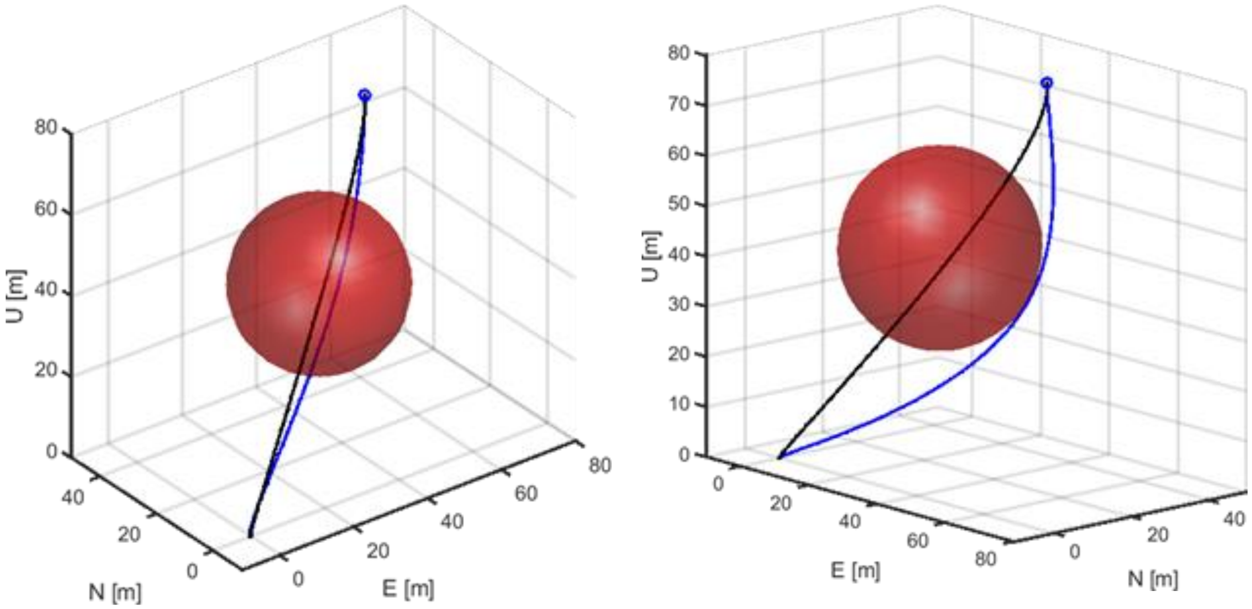


Figure A-28 Automated Landing Scenario 1: Single Obstacle

A more complex example is shown in Figure A-29 where multiple obstacles appear in the space. The vehicle has an initial position at (100, 100, 80), no initial velocity, and must land at the origin, i.e. landing site, with zero final velocity and acceleration. Front and rear isometric views are shown in the figure. The path planner successfully finds a path adhering to all constraints while minimizing the acceleration profile. The same obstacle scenario is repeated in Figure A-30 but in this scenario the vehicle has non-zero initial velocity flying a vector to the origin. The trajectory planner accounts for the initial velocity and the path goes above the first set of obstacles and then weaves between the latter ones on the approach to the landing site.

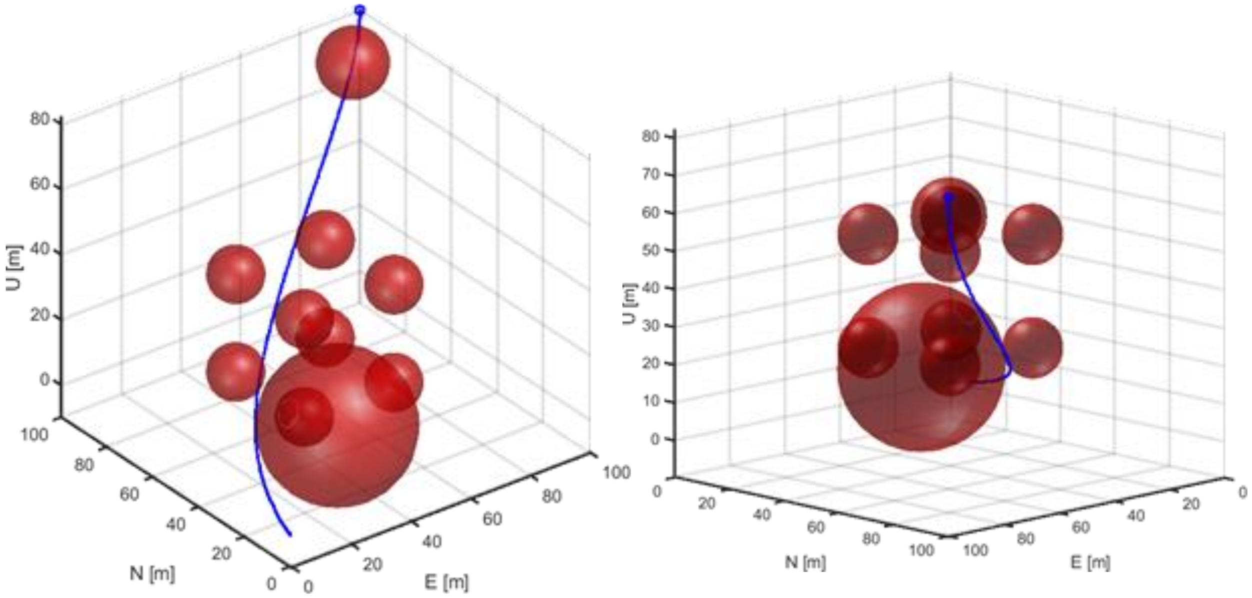


Figure A-29 Automated Landing Scenario 2: Multiple Obstacles, No Initial Velocity

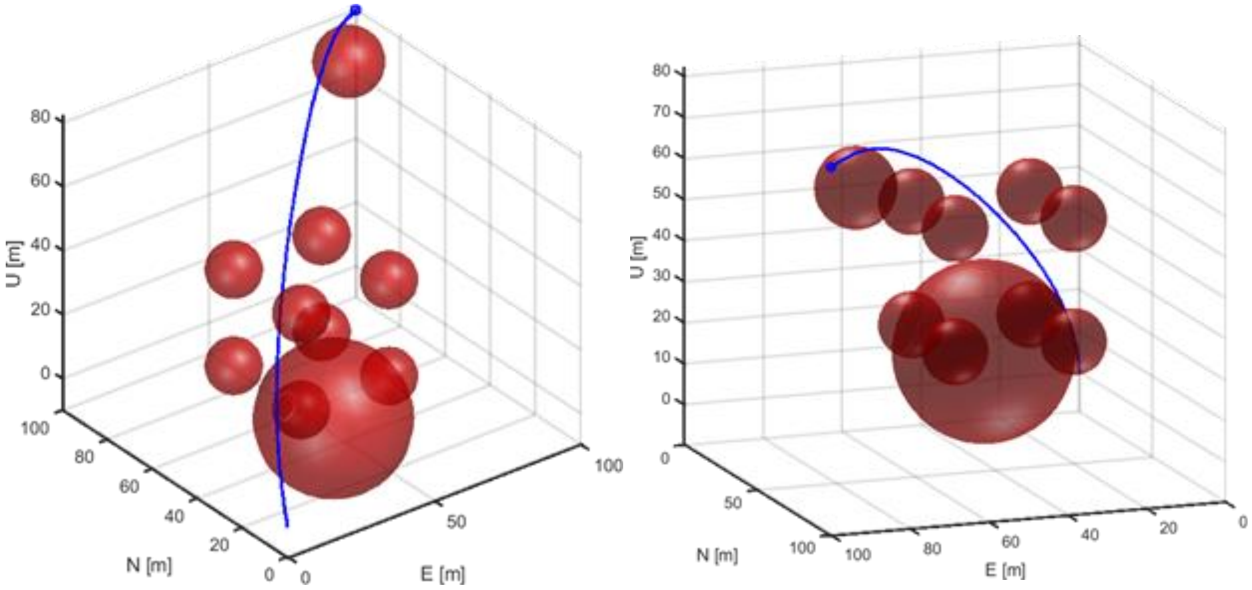


Figure A-30 Automated Landing Scenario 2: Multiple Obstacles with Initial Velocity

A3.2.3 Develop RTA Monitor Function Requirements

In this section, a modified set of requirements for an alternative development approach that relies on RTA monitoring and switching behavior is developed. A diagram of the modified algorithm structure is shown in Figure A-31. (To simplify the diagram, feedback loops of aircraft sensor outputs and state estimates to each of the other blocks have been omitted and the pilot command path has been renamed Constant Neutral Command). This design duplicates the TRC, ACAH and mixing components, adding a constant command input to the lower path. It also adds an RTA Monitor that continually checks the aircraft state, and a switch used to disconnect the autoland path in favor of the neutral input path.

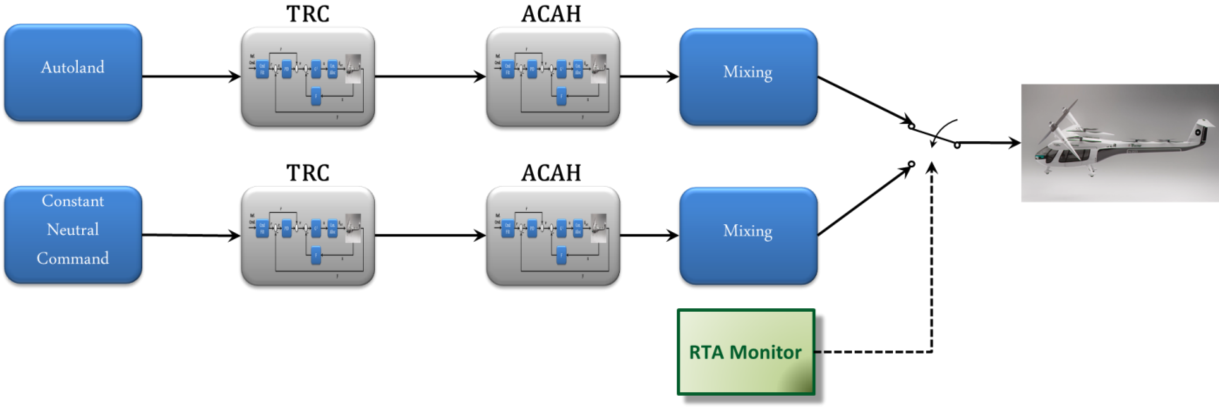


Figure A-31 RTA Variant of the Autoland Algorithm Structure

The RTA monitor looks for impending occurrence of every hazard associated with the functions allocated to the autoland component. If such a hazard appears imminent, the RTA monitor switches to the neutral input control path. From the discussion in Section A3.2, the TRC component in that path will attempt to achieve a position-hold, commanding aircraft attitude as appropriate to do so from the current aircraft state. At this point, the onboard human pilot can take control, using the inceptors to direct the aircraft to a landing.

This design incorporates automated monitoring for a potentially large number of hazardous conditions, many of which may have an onset that occurs so rapidly that an onboard human pilot may not detect them until it is too late to salvage the aircraft. By providing a lower, neutral command path, the design avoids the potential for automation surprise that may occur for a pilot who lacks situational awareness and finds themself in control of a complex aircraft in a near-hazard condition.

A3.3 Identify System Level Safety Objectives (SFHA)

The system level safety objectives are established by an analysis of the planned system functions to be implemented using the functional hazard assessment (FHA) methodology described in ARP4761 [23]. For the study example, an extract of the eCRM-001 flight-propulsion control system FHA is provided in Figure A-32, Example Artifact 2, eCRM-SFHA.

Figure A-32 Example Artifact 2: eCRM SFHA

BAART-ECRM-SFHA

Flight-Propulsion Control System Functional Hazard Assessment (FHA) for the eCRM-001 Airplane

SIZE	FSCM NO	DWG NO	REV
A		eCRMSFHA-100	-
SCALE		SHEET	1 of 11

REVISIONS				
CN NO.	REV	DESCRIPTION	DATE	APPROVED
		Initial Release	15Jan2020	

1. Introduction

The assessment captured herein represents the system level functional hazard assessment (SFHA) for the eCRM-001 Flight-Propulsion Control System (FPCS).

1.1. References

[1] eCRM-001 Airplane Design Requirements Document

[2] eCRM-001 Flight-Propulsion Control System Requirements Document

[3] 14CFR/CS Part 23, Amendment 23-64

[4] ASTM F3230-17, “Safety Assessment of Systems and Equipment in Small Aircraft”

[5] ASTM F3061-17, “Standard Specification for Systems and Equipment in Small Aircraft”

[6] ARP4761 “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Aircraft Systems and Equipment”

[7] eCRMAFHA-100, Aircraft Functional Hazard Assessment for the eCRM-001 Airplane

Editor’s Note: Document reference numbering within an example artifact will be to the documents listed as references in this section rather than the overall report reference list.

1.2. Glossary

This section captures specific terms and definitions used within the SFHA.

Term	Definition
Uncommanded	Activation of a function without pilot command input or erroneously activated due to equipment failure.
Minimum Acceptable Control (MAC)	An airplane configuration under which the normal acceptable control performance criteria will still be satisfied and when lost will result in the failure condition effects described.

2. System Description Summary

Editor’s note: Duplicate system description summary removed for brevity.

3. Flight-Propulsion System SFHA Development

3.1. SFHA Inputs

A review of the Flight-Propulsion Requirements Document [2] identified the system level functions captured in Table eCRM-001 FPCS Function List. These functions will be the subject of the safety evaluation herein.

Table eCRM-001 FPCS Function List

1. Control Lift for VTOL
   1.1. Control left/right wingtip rotor lift
   1.2. Control left/right wing rotor stack lift
   1.3. Control forward/aft rotor stack lift
2. Control Yaw - VTOL
3. Control Roll – VTOL
4. Control Pitch – VTOL
5. Etc.

3.2. Review & Confirm System Level Functions

Editors' Note: Not included in this example for brevity.

3.3. Determine Failure Conditions

Editors' Note: Only the "Control Lift for VTOL" system function is developed in this example. All other system functions, at the level of the functional breakdown defined in 4.1, would be developed in a similar fashion.

3.3.1. Failure Condition Identification Matrix

A failure condition identification matrix was constructed for the function "*Control Lift for VTOL*". This initial matrix is presented in Table SFHA-1 eCRM-001 System Failure Condition Identification Matrix. Postulated failure condition descriptions are captured for Total Loss of function, Partial Loss of function and Malfunction (erroneous operation) of function.

3.3.2. Crew Awareness

Editors' Note: Not included in this example for brevity.

3.4. Assess Failure Condition Effects

The effects of each of the identified failure condition on the aircraft, flight crew and occupants other than the flight crew have been assessed. The effects are captured based on their immediate effect on aircraft, flight crew and occupants during the phase of flight being analyzed.

The captured effects of each failure condition are shown in column (5) of the SFHA worksheet tables.

3.4.1. eCRM-001 Flight Phases

Editors' Note: This section is identical to the flight phase description provided in the AFHA example and not included here for brevity.

3.4.2. Operational Conditions

Editors' Note: Not included in this example for brevity.

3.4.3. Environmental Conditions

Editors' Note: Not included in this example for brevity.

3.5. Classify Failure Conditions Based on Effect Severity

Editors' Note: This section is identical to the information provided in the AFHA example and not included here for brevity.

The classification of each failure condition for each flight phase is captured in Column (6) of the SFHA worksheet tables.

Table SFHA-1 eCRM-001 System Failure Condition Identification Matrix

ID #	Aircraft Function	Total Loss	Partial Loss	Malfunction
1	Control Lift for VTOL			
1.1	Provide wingtip rotor lift			
1.1.1	Provide wingtip rotor lift	**1.1.1.TL** Loss of wingtip rotor lift	**1.1.1.PL** Partial loss of wingtip rotor lift	**1.1.1.MF1** Uncommanded wingtip rotor lift – single rotor **1.1.1.MF2** Uncommanded wingtip rotor lift – dual rotor **1.1.1.MF3** Erroneous wingtip rotor lift intensity – excessive **1.1.1.MF4** Erroneous wingtip rotor intensity - diminished
1.2	Provide wing lifting rotor stack lift			
1.2.1	Provide wing lifting rotor stack lift	**1.2.1.TL** Loss of wing lifting rotor stack lift	**1.2.1.PL** Partial loss of wing lifting rotor stack lift	**1.2.1.MF1** Uncommanded wing lifting rotor stack lift – single rotor **1.2.1.MF2** Uncommanded wing lifting rotor stack lift – dual rotor **1.2.1.MF3** Erroneous wing lifting rotor stack lift intensity – excessive **1.2.1.MF4** Erroneous wing lifting rotor stack intensity - diminished
1.3	Provide forward/aft lifting rotors lift			
1.3.1	Provide fuselage lifting rotor stack lift	**1.3.1.TL** Loss of fuselage rotor stack lift	**1.3.1.PL** Partial loss of fuselage rotor stack lift	**1.3.1.MF1** Uncommanded fuselage rotor stack lift – single rotor **1.3.1.MF2** Uncommanded fuselage rotor stack lift – dual rotor **1.3.1.MF3** Erroneous lift fuselage rotor stack lift intensity – excessive **1.3.1.MF4** Erroneous lift fuselage rotor stack intensity - diminished
Etc.				

3.6. SFHA Assumptions and Hazard Classification Criteria

Assumptions made while accomplishing the effect evaluation of each failure condition have been captured and numerically identified for reference. Table SFHA-2 SFHA Assumptions (SSMP)/Notes presents the analysis assumptions and notes criteria that made during the development of this functional hazard assessment.

Table SFHA-2 SFHA Assumptions (SSMP)/Notes

Assumption Identifier	**Description**
SSMP 1.1.1-1	Loss of full wingtip rotor lift performance (i.e.100% of required lift) does not cause loss of minimum acceptable lift capability ([7] AFHA ASMP 2.1.1-1)
SSMP 1.1.1-2	Each wingtip rotor is capable of providing full lift performance (i.e.100% of required lift) when operating on single electric motor.
SSMP 1.1.1-3	Full wingtip rotor takeoff/land lift performance capability (100% of required lift) is lost with loss of rotor blade pitch control, loss of both electric motors, mechanical failure of summing gearbox or loss of ability to tilt nacelle to vertical position (90 degrees).
SSMP 1.1.1-4	Partial wingtip rotor takeoff/land lift performance capability (50% of required lift) occurs when rotor blade pitch is at mid-range position (i.e. between feather and full blade angle capability)
SSMP 1.2.1-1	Loss of full wing rotor stack lift performance (i.e.100% of required lift) does not cause loss of minimum acceptable lift capability (([7] AFHA ASMP 2.1.1-1)
SSMP 1.2.1-2	Full wing rotor stack takeoff/land lift capability (100% of required lift) is lost with loss of both redundant electric motors or erroneous rotor stack stow.
SSMP 1.2.1-3	Partial wing rotor stack takeoff/land lift performance capability (50% of required lift) occurs with loss of single electric motor or single rotor.
SSMP 1.3.1-1	Loss of full fuselage rotor stack lift performance capability (i.e.100% of required lift) does not cause loss of minimum acceptable lift capability (([7] AFHA ASMP 2.1.1-1)
SSMP 1.3.1-2	Full fuselage rotor stack takeoff/land lift performance capability (100% of required lift) occurs with the loss of both redundant electric motors or erroneous rotor stack stow.
SSMP 1.3.1-3	Partial fuselage rotor stack takeoff/land lift performance capability (50% of required lift) occurs with loss of single electric motor or single rotor.

4. SFHA Output Summary

This section summarizes the failure effects and their associated severity classifications. For details associated with each failure conditions see the populated worksheets in the appendix.

Table SFHA-3 FPCS Catastrophic & Hazardous FC Summary

1	2	3	4	6
Function	**FC #**	**Failure Condition (Hazard Description)**	**Flight Phase**	**Classification**
Control Lift for VTOL: Provide wingtip rotor lift	1.1.1.MF2	Uncommanded dual wingtip rotor provided vertical lift	T1, T2, T3, L1, L2	Hazardous
	1.1.1.MF3	Uncommanded wingtip rotor provided vertical lift (left or right) – excessive lift	T1, T2, T3, L1, L2	Hazardous
	1.1.1.MF4	Uncommanded wingtip rotor provided vertical lift (left or right) – diminished lift	T1, T2, T3, L1, L2	Hazardous
Etc.				

Editor's Note: Only failure conditions developed in the example were summarized in Table FPCS Catastrophic & Hazardous FC Summary rather than all system catastrophic and hazardous conditions.

Table SFHA-4 eCRM-001 FPCS Functional Hazard Assessment Worksheets (excerpt)

FUNCTIONAL HAZARD ASSESSMENT						
System:	**Flight-Propulsion Control System**					**Rev Date:** **10 December 2019**
Function:	**Control Lift for VTOL**					
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
1.1.1.TL	Loss of all (100%) single wingtip rotor provided vertical lift (left or right)	T1, T2, T3, L1, L2	A) Large reduction in lift capability. No remaining lift capability safety margin. B) Slight increase in Pilot workload to identify and compensate for decreased lift performance. Pilot performs rejected takeoff or continues normal landing. C) Inconvenience for passengers.	III	SSA	SSMP 1.1.1-1 SSMP 1.1.1-2 SSMP 1.1.1-3
1.1.1.PL	Loss of 50% of single wingtip rotor provided vertical lift(left or right)	T1, T2, T3, L1, L2	A) Significant reduction in lift capability. Airplane lift rate underperforms commanded values resulting decreased lift performance. B) Slight increase in Pilot workload to identify and compensate for decreased lift performance. C) No effect.	IV	SSA	SSMP 1.1.1-4
1.1.1.MF1	Uncommanded single wingtip rotor provided vertical lift (left or right)	T1, T2, T3, L1, L2	A) Slight asymmetric lift results in induced airplane roll response. B) Pilot able to maintain control using increase lift on wingtip rotor opposite failed rotor; Pilot adjusts electric motor rpms on failed wingtip. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	III	SSA	
1.1.1.MF2	Uncommanded dual wingtip rotor provided vertical lift (left or right)	T1, T2, T3, L1, L2	A) Airplane unstable along desired flight path. Asymmetric lift results in induced airplane roll response. B) Pilot able to maintain roll control using increase lift on wingtip rotor opposite failed rotor. Pilot adjusts lift on other rotors for immediate landing. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	II	SSA	

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

CLASS I —	CATASTROPHIC
CLASS II	HAZARDOUS
CLASS III	MAJOR
CLASS IV	MINOR
CLASS V	NO EFFECT

FUNCTIONAL HAZARD ASSESSMENT

System:	**Flight-Propulsion Control System**					**Rev Date:** **10 December 2019**
Function:	**Control Lift for VTOL**					
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
1.1.1.MF3	Uncommanded wingtip rotor provided vertical lift (left or right) – excessive lift	T1, T2, T3, L1, L2	A) Airplane unstable along desired flight path. Excessive lift on single wingtip causes slight induced airplane roll. B) Pilot able to maintain control using increase lift on wingtip rotor opposite failed rotor; reduce electric motor rpms on failed wingtip. Pilot adjusts lift on other rotors for immediate landing. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	II	SSA	
1.1.1.MF4	Uncommanded wingtip rotor provided vertical lift (left or right) – diminished lift	T1, T2, T3, L1, L2	A) Airplane unstable along desired flight path. Diminished lift on single wingtip causes slight induced airplane roll. B) Pilot able to maintain control using decreased lift on wingtip rotor opposite failed rotor; increase electric motor rpms on failed wingtip. Pilot adjusts lift on other rotors for immediate landing. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	II	SSA	
1.2.1.TL	Loss of 100% of single wing rotor stack provided vertical lift (left or right)	T1, T2, T3, L1, L2	A) Large reduction in lift capability. No remaining lift capability safety margin. B) Slight increase in Pilot workload to identify and compensate for decreased lift performance. Pilot performs rejected takeoff or continues normal landing. C) Inconvenience for passengers.	III	SSA	SSMP 1.2.1-1 SSMP 1.2.1-2

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

• CLASS I	—	CATASTROPHIC
• CLASS II		HAZARDOUS
• CLASS III		MAJOR
• CLASS IV		MINOR
• CLASS V		NO EFFECT

FUNCTIONAL HAZARD ASSESSMENT

System:	**Flight-Propulsion Control System**					**Rev Date:** **10 December 2019**
Function:	**Control Lift for VTOL**					
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
1.2.1.PL	Loss of 50% of single wing rotor stack provided vertical lift (left or right)	T1, T2, T3, L1, L2	A) Slight reduction in lift capability. Airplane climb rate underperforms commanded values resulting decreased climb performance. B) Slight increase in Pilot workload to identify and compensate for decreased lift performance. C) No effect.	IV	SSA	SSMP 1.2.1-3
1.2.1.MF1	Uncommanded wing rotor stack provided vertical lift (left or right) – single rotor	T1, T2, T3, L1, L2	A) Erroneous lift on one stack results in asymmetric lift which induces slight airplane roll response. B) Pilot maintains control using increase lift on wing rotor stack opposite failed rotor; Pilot adjusts electric motor rpms on failed wing rotor stack. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	III	SSA	
1.2.1.MF2	Uncommanded wing rotor stack provided vertical lift (left or right) – dual rotor	T1, T2, T3, L1, L2	A) Asymmetric lift condition induces airplane roll response. B) Pilot maintains roll control using increase lift on wing rotor stack opposite failed rotor. Pilot adjusts lift on other rotors for immediate landing. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	III	SSA	
1.2.1.MF3	Uncommanded wing rotor stack provided vertical lift (left or right) – excessive lift	T1, T2, T3, L1, L2	A) Asymmetric lift condition may cause induced airplane roll. B) Pilot maintains control using increase lift on wing rotor stack opposite failed rotor; Pilot adjusts electric motor rpms on failed wing rotor stack. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	IV	SSA	

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

• CLASS I —	CATASTROPHIC
• CLASS II	HAZARDOUS
• CLASS III	MAJOR
• CLASS IV	MINOR
• CLASS V	NO EFFECT

FUNCTIONAL HAZARD ASSESSMENT

System:	**Flight-Propulsion Control System**					**Rev Date:** **10 December 2019**
Function:	**Control Lift for VTOL**					
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
1.2.1.MF4	Uncommanded wing rotor stack provided vertical lift (left or right) – diminished lift	T1, T2, T3, L1, L2	A) Asymmetric lift condition may cause induced airplane roll. B) Pilot maintains control decreasing lift on wing rotor stack opposite failed rotor; Pilot adjusts electric motor rpms on failed wing rotor stack. C) Passenger experience discomfort due to differential lift induced roll maneuvers.	IV	SSA	
1.3.1.TL	Loss of 100% of single fuselage rotor stack provided vertical lift (forward or aft)	T1, T2, T3, L1, L2	A) Large reduction in lift capability. No remaining lift capability safety margin. B) Slight increase in Pilot workload to identify and compensate for decreased lift performance. Pilot performs rejected takeoff or continues normal landing. C) Inconvenience for passengers.	III	SSA	SSMP 1.3.1-1 SSMP 1.3.1-2
1.3.1.PL	Loss of 50% of single fuselage rotor stack provided vertical lift (forward or aft)	T1, T2, T3, L1, L2	A) Slight reduction in lift capability. Airplane climb rate underperforms commanded values resulting decreased climb performance. B) Slight increase in Flight Crew workload to identify and compensate for decreased lift performance. C) No effect.	IV	SSA	SSMP 1.3.1-3
1.3.1.MF1	Uncommanded fuselage rotor stack provided vertical lift (forward or aft) – single rotor	T1, T2, T3, L1, L2	A) Slight increase in lift provided by fuselage rotor stack. B) Pilot maintains control through adjustments to other lifting rotors. C) No effect.	IV	SSA	

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

• CLASS I	—	CATASTROPHIC
• CLASS II		HAZARDOUS
• CLASS III		MAJOR
• CLASS IV		MINOR
• CLASS V		NO EFFECT

FUNCTIONAL HAZARD ASSESSMENT						
System:	**Flight-Propulsion Control System**					**Rev Date:** **10 December 2019**
Function:	**Control Lift for VTOL**					
1	**2**	**3**	**4**	**5**	**6**	**7**
Ref. No.	**Failure Condition** ***(Hazard Description)***	**Flight Phase**	**Effect of Failure Condition on:** **A) Aircraft,** **B) Crew,** **C) Occupants**	**FC Class**	**Cert Approach**	**Remarks / Justification**
1.3.1.MF2	Uncommanded fuselage rotor stack provided vertical lift (forward or aft) – dual rotor	T1, T2, T3, L1, L2	A) Increase in lift provided by fuselage rotor stack. B) Pilot maintains control through adjustments to other lifting rotors. C) No effect.	III	SSA	
1.3.1.MF3	Uncommanded fuselage rotor stack provided vertical lift (forward or aft) – excessive lift	T1, T2, T3, L1, L2	A) Slight increase in lift provided by fuselage rotor stack. B). Pilot maintains control through adjustments to other lifting rotors C) No effect.	IV	SSA	
1.3.1.MF4	Uncommanded fuselage rotor stack provided vertical lift (forward or aft) – diminished lift	T1, T2, T3, L1, L2	A) Slight decrease in lift provided by fuselage rotor stack. B). Pilot maintains control through adjustments to other lifting rotors C) No effect.	IV	SSA	
Etc.						

OPERATIONAL FLIGHT PHASES (Col. 3)

GROUND	**TAKEOFF**	**INFLIGHT**		**LANDING**
G1: Taxi	T1: Break ground to Hover	F1: Climb	F4: Go-Around	L1: Transition-Fwd to Hover
	T2: Transition – Hover to Fwd	F2: Cruise	F5:	L2: Hover Descend to ground
	T3: Rejected Takeoff	F3: Descent	F6:	

HAZARD CLASSIFICATIONS (Col. 5)

- CLASS I — CATASTROPHIC
- CLASS II HAZARDOUS
- CLASS III MAJOR
- CLASS IV MINOR
- CLASS V NO EFFECT

--------End of System FHA Excerpt-------

A3.4 Develop System Architecture

This section develops the architecture strategies for the baseline development approach and the RTA architecture approach.

A.3.4.1 Develop Baseline Architecture

The high integrity baseline self-monitoring architecture concept is presented in Figure A-33. Two high confidence functional control applications are executing in two independent computer platforms and the resulting computation results are compared. When the calculations match, the computation result is output on the digital busses to the effectors. If the functional computations disagree, the FCC relinquishes control to the alternate FCC.

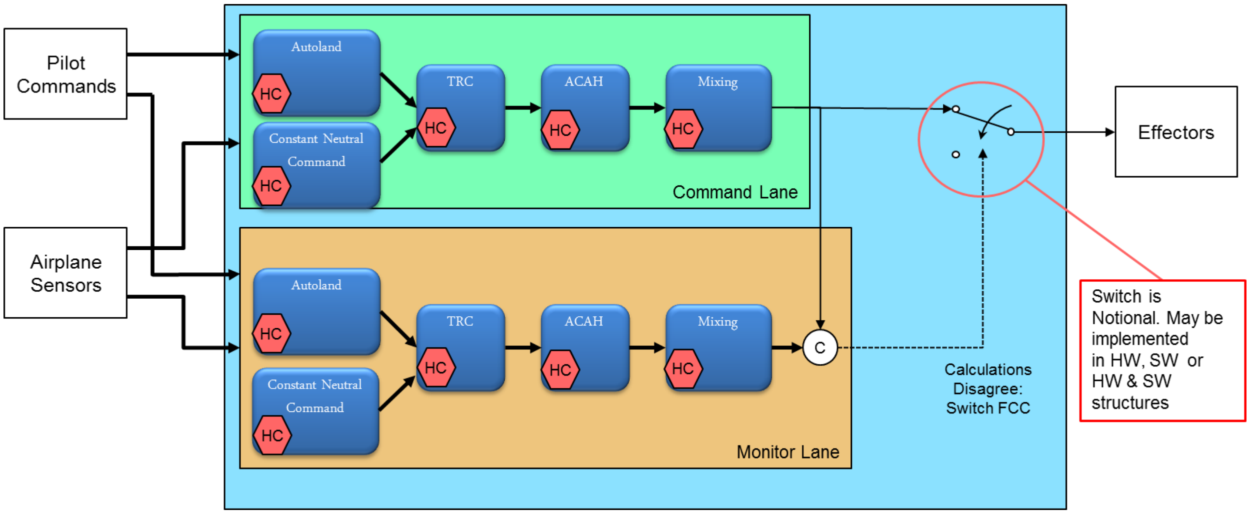


Figure A-33 Baseline Control Architecture

In the baseline scenario, the Autoland, TRC, ACAH, and mixing components are all high-confidence systems.

A3.4.2 Develop RTA Architecture Criteria

The RTA architecture approach allocates the functions so as to take advantage of a high-integrity, high confidence monitoring mechanism to mitigate development errors as well as implementation mistakes. Figure A-34 presents this high level concept.

A high-confidence Constant Neutral control path is redundantly implemented in both Command and Monitor Lanes. The results of these calculations are used within the RTA architecture for two purposes; 1) To provide pilot reversionary control capability in the event of an RTA Monitor exceedance; 2) Establish that the Command and Monitor Lane physical implementations have reached the same calculated results (establish FCC integrity).

An automated low-confidence control path provides normal vehicle controlled operation. This low-confidence Command Lane path is independently monitored by a high-confidence RTA Monitor path. The Autoland downstream TRC-ACAH-Mixing elements may be identical to the Constant Neutral path (as shown in Figure A-34) or integrated in to the Autoland low confidence calculations

Editor's Note: The switches shown in Figure A-34 are notional. These may be physically implemented in electronic hardware, software or a combination.

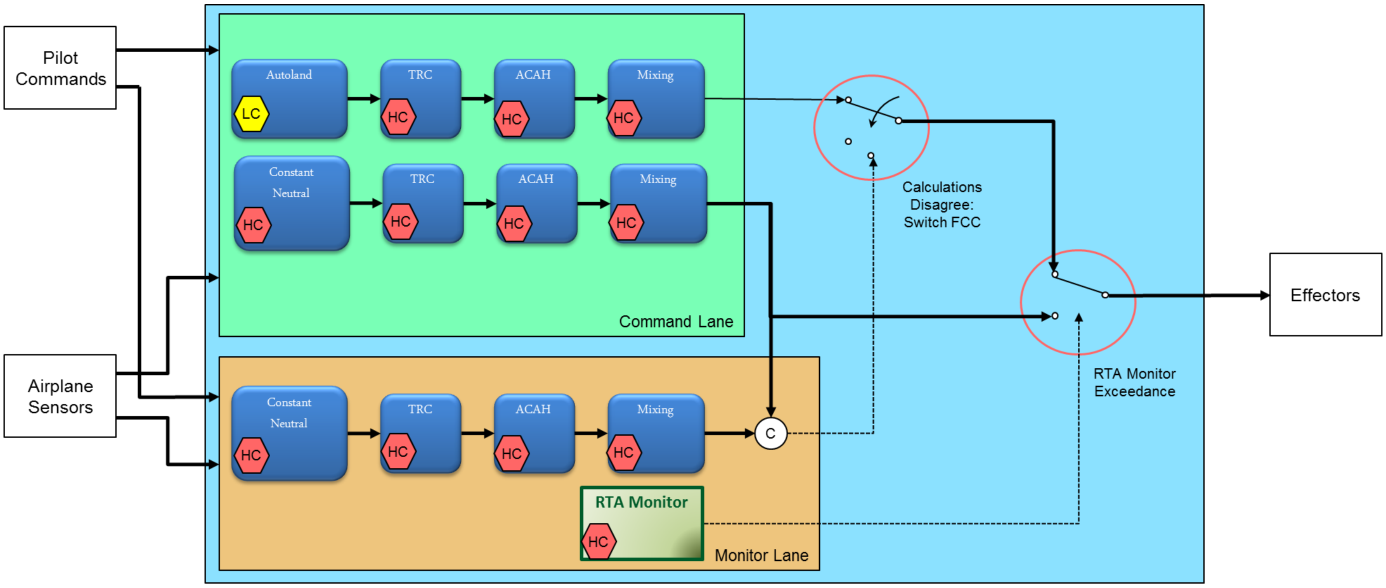


Figure A-34 RTA Allocated Autoland Architecture

The central idea of the RTA approach is to ensure system safety and correct operation by validating a command signal from a low-confidence (LC), or untrusted, system. If the signal from the LC system cannot be validated, i.e., it could drive the system into a defined *unsafe* region, then the RTA mechanism blocks the output from the LC system and outputs the signal from a reversionary, high-confidence (HC) system. The HC system has been certified during design-time and follows a traditional V&V process. In this manner the output of the RTA system is a validated signal that produces quantifiably safe behavior regardless of the operating scenario or time.

To achieve this, behavior formal definitions for system safety and regions defining a *safe state-space*, or domain, must be defined at design-time. In prior work, Barron Associates outlined a process for constructing a switching condition that can be implemented in the RTA monitor and switch mechanism to ensure that the aircraft does not enter an unsafe state [43]. This process requires designers to define a nested sequence of aircraft states that satisfy increasingly stringent notions of safety, as depicted in Figure A-35.

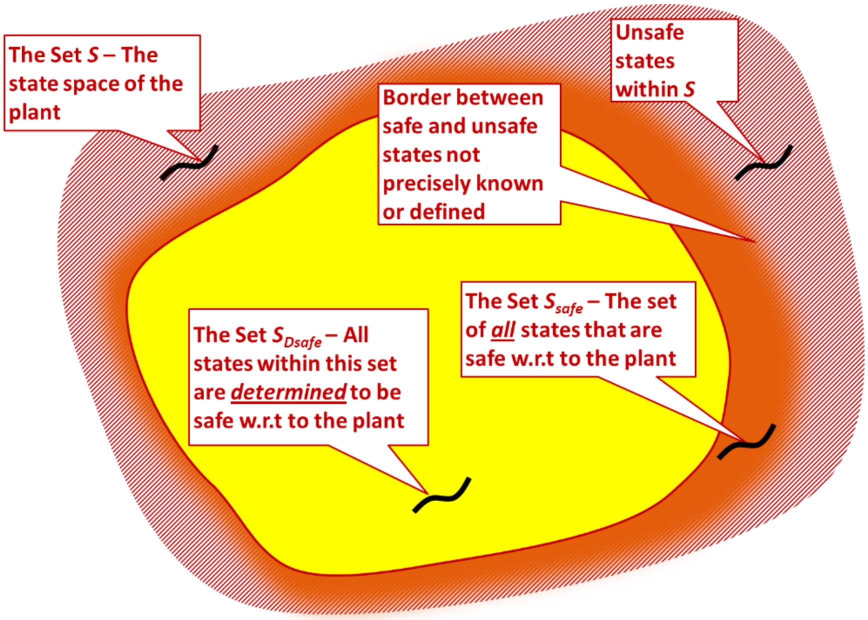

The Set S – The state space of the plant
Border between safe and unsafe states not precisely known or defined
Unsafe states within S
The Set S_{safe} – The set of *all* states that are safe w.r.t to the plant
The Set S_{Dsafe} – All states within this set are *determined* to be safe w.r.t to the plant

Figure A-35 Nested State Definitions for Levels of Safety

This process is summarized in the following steps:

1. **Define set S**: The set S is the set of all states of the system function. This step requires designers to determine the collection of all variables related to the aircraft that play a role in its safety. That is, it includes not just the states that may be used to build a dynamic simulation model, but also structural states, power plant operation, and other plant configuration states and environmental states.
2. **Define the set $\boldsymbol{S_{safe}}$**: The set $S_{safe} \subseteq S$ is the set of all states that are ***safe*** with respect to the system function. Safe states are operating points in which the system functions as intended. This step requires designers to explicitly determine what it means for the system to function "as intended" and how it relates to system state. If the plant is a physical system, then states within the set will not cause or lead to:
    a. environmental conditions that cause uncontrollable or upset conditions;
    b. physical damage of the plant itself;
    c. damage or adverse conditions to other plants or systems within the plant's influence; or,
    d. harm or injury to human operators or other persons.

3. **Construct the set $\boldsymbol{S_{Dsafe}}$** : The set $S_{Dsafe} \subseteq S_{safe}$ is the set of all states determined to be safe with respect to the system function. Note that it is often the case that the set S_{safe} cannot be precisely determined or defined. That is, the exact border separating safe from unsafe states is typically uncertain due to modeling errors, inaccuracies in measuring or estimating system states, changing environmental conditions, etc. For this reason, designers typically add in safety margins to account for such uncertainties; these safety margins are often a matter of judgment. The set S_{Dsafe} accounts for these margins and comprises the set of states that, by definition, the aircraft must never depart. Keeping the aircraft within this safety set is the primary goal of the RTA monitor and switch mechanism. As such, a point $\mathbf{x_o} \in \mathbf{S_{Dsafe}}$ is said to be Type I safe. Figure A-35 gives a graphical illustration of the sets S, S_{safe}, and S_{Dsafe}. Note that these sets are functions only of the aircraft system function and are not dependent on specific algorithms or their corresponding implementations.

4. **Design Recovery Operations**: Determine the recovery operation that the backup control algorithm will perform if the LC system is switched out by the RTA monitor and switch mechanism. Specify this recovery operation in terms of a set $Q \in S_{Dsafe}$ and a time interval *T* having the property that, following a switch from the main controller to the backup controller, the backup controller will attempt to drive the system being controlled into a state inside *Q* within a time interval *T > 0*. The quantities *Q* and *T* may be dependent on the state of the aircraft system at the time the switching activity occurs. We define the region *Q* to be that set of states at which the *off-nominal condition* (which triggered the RTA monitor/switch) is resolved or alleviated; it need not be the final state region in the recovery process. Once the state reaches *Q*, then it will be in a stable, safe region of attraction. See Figure A-36 for an illustration of these states.

5. **Determine the Type II Safety Region**: The Type II safety region is the subset of states in S_{Dsafe} such that, upon switching to the backup controller, the state trajectory can converge to at least one point in *Q* within the desired time *T*, and that trajectory is entirely contained within S_{Dsafe}. Type II safety accounts for the behavior of the combined system consisting of the plant and the reversionary system. Its definition is such that, from any point in the Type II safe region, the RTA switch could activate the reversionary system, which can then maintain control over the plant, driving it to a desired region, *Q,* in the state space in a finite amount of time *T*. Physical systems have momentum, so state trajectories require a finite amount of time to be altered and may also exhibit overshoot. Further, control mode switches often result in transient behavior. By definition, Type II safety requires that any resulting overshoot or transients involved during the switching process and transition to *Q* will not compromise plant safety, i.e., will not leave the set S_{Dsafe}.

6. **Determine Time Horizon**: Determine τ, the time horizon used for planning purposes by the RTA monitor and switch mechanism. In most cases, the RTA, LC, and HC systems will be implemented as discrete-time system that repeatedly check the safety properties of the current aircraft system state, then make a decision about whether or not to switch to the HC controller. The quantity τ must be greater than the maximum elapsed time between these successive checks.

7. **Determine Type III Safety Region**: The Type III safety region is the set of states that are Type II safe and that have the property that every possible output of the main controller for a time period τ results in a state trajectory entirely contained within the Type II safe region. From any point in the Type III safe region, the RTA switch can pass any commands from the advanced system to the plant for at least τ seconds without exiting Type II safety. The Type III safety region then defines the boundary for the RTA switching mechanism.

After these determinations have been made, the RTA monitor and switch mechanism can be designed to switch from the main controller to the backup controller any time the aircraft state is found to be outside the Type III safety region. The Type I, II and III safety regions are illustrated in Figure A-36.

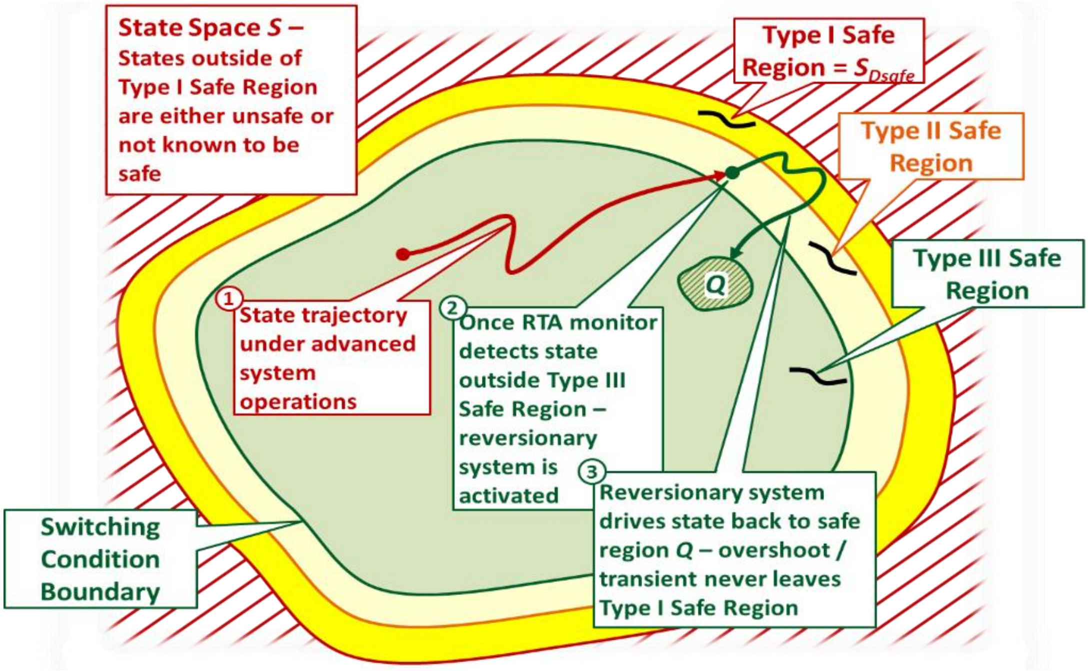

State Space S – States outside of Type I Safe Region are either unsafe or not known to be safe
Type I Safe Region = S_{Dsafe}
Type II Safe Region
Type III Safe Region
Q
1 State trajectory under advanced system operations
2 Once RTA monitor detects state outside Type III Safe Region – reversionary system is activated
3 Reversionary system drives state back to safe region Q – overshoot / transient never leaves Type I Safe Region
Switching Condition Boundary

Figure A-36 Type I, II & III Safety Regions

These definitions are perhaps more readily understood from the perspective of the RTA monitor's decision logic. The RTA monitor and switch component must guarantee that the aircraft never leaves the Type I safe region, which ensures that the aircraft continues to operate safely. To guarantee this, every τ seconds, it checks to see if the current state is inside the Type III safety region. If the state is in this region, the RTA monitor can allow the advanced system's commands to pass through to the plant. In the worst case, on its next check τ seconds later, the plant's state will have transitioned out of the Type III region, but it will not have transitioned beyond the Type II safe region (part of the definition of Type III safety). In that case, the RTA component must switch to the reversionary controller. Because the aircraft is now in the Type II safe region, the resulting transient will never leave the Type I safe region (part of the Type II safety definition) and the reversionary system can successfully maintain safe control/management of the aircraft.

A3.5 Derive System Requirements

The baseline system develops FPCS functional requirements per the conventional system development paradigm. Airplane requirements are decomposed and allocated to system requirements with design decisions creating derived requirements.

Editor's Note: The development of baseline functional requirements has been omitted in the interest of example brevity since this has no effect on study results.

A3.5.1 Derive RTA System Requirements

Requirements of the RTA monitor and switching components from Figure A-34 are largely determined based on the safety and performance requirements of the high-confidence (neutral input/TRC/ACAH) path. That is, the RTA components must ensure that the aircraft state remains within an operating envelope from which the high-confidence control path can ensure safety of the aircraft. If, at any time while the low-confidence (autoland) path is in control, there is a significant risk that the aircraft state may cross that boundary, the RTA components must switch to the high-confidence path.

The process for RTA development that was outlined in A3.4.2 is applied to an example scenario involving the eCRM-001 vehicle in the subsections below. These subsections address the following:

1. Determine the intended recovery behavior of the high-confidence control path after the RTA components switch to it.
2. Determine the set S_{Dsafe} of aircraft states that constitute the operational envelope of the high-confidence path. Of special interest is the boundary, called the Type I Safety Boundary, between this set and the states that are outside the envelope.
3. Determine the subset of S_{Dsafe} from which the high-confidence control path will correctly execute the intended recovery behavior while remaining inside the set S_{safe}. This subset is called the Type II Safety Region, and its boundary is called the Type II Safety Boundary. The RTA components must ensure that the low-confidence control path never takes the aircraft to a state that is outside this boundary.
4. Determine the time horizon τ used for planning purposes in the RTA design.
5. Determine the subset within the Type II Safety Region from which the low-confidence control path cannot force the aircraft to exit the Type II Safety Region during the update period of the RTA components. This subset is called the Type III Safety Region, and its boundary is called the Type III Safety Boundary. This boundary acts as a conservative buffer to account for the non-instantaneous, discrete-time operation of the RTA components.

Once these safety boundaries have been determined, behavior of the RTA monitor components is straightforward:

- If the RTA monitor determines that the low-confidence control path has driven the aircraft beyond the Type III Safety Boundary, switch to the high-confidence path.

The Type I, Type II, and Type III Safety Boundaries are defined relative to one another in sequence. Processes to derive each of them for the example system are described below.

Intended Recovery Behavior Determination

The Type I, II, and III Safety Boundaries are defined relative to the intended recovery behavior of the high-confidence control path. These recovery requirements are defined in terms of a set of aircraft states *Q* and a maximum time *T*. Specifically, when the high-confidence control path is engaged, it will attempt to drive the vehicle state into the set *Q* within some time limit *T*.

Recall that for the eVTOL example, the high-confidence control path consists of a TRC-ACAH-Mixing sequence of modules, with a neutral input to the TRC controller. This control path will thus attempt to hold the aircraft at a zero-velocity state. Therefore, the set *Q* can be defined as the set of vehicle states corresponding to a zero aircraft velocity, in which its motion in all directions has been arrested.

There is considerable flexibility in choosing the quantity *T*, but its value has a number of downstream effects. A small value of *T* will force the high-confidence control path to accelerate aggressively to arrest a high-velocity landing operation, whereas a larger value of *T* will permit more moderate responses. Thus, larger values may be preferable from the perspective of passenger comfort and structural loading. On the other hand, small values of *T* will generally result in greater freedom for the low-confidence control path to perform its function, since the high-confidence control path is able to quickly recover from any potentially unsafe situation.

For the purposes of this example, we define *Q* as the set of vehicle states with zero velocity, which corresponds to a near stationary hover. We define *T*=2s as the maximum amount time allowed for the high-confidence path to achieve zero velocity, starting from any initial velocity.

Type I Safety Boundary Determination

As previously described, the Type I Safety Boundary is the limit of the defined safe operating envelope for the high-confidence control path. It equals the intersection of two sets:

1. The set of states from which the high-confidence control path can execute its recovery operation (i.e., drive the vehicle state into the set Q within time T); and
2. The set of states in which the aircraft itself is determined to be safe with respect to its various functions (i.e., is within its physical limits, is stable, etc.).

For the eVTOL case study application, we note that the TRC, ACAH, and mixing components of the high-confidence control path are also used by the human pilot when the autoland function is not active. Thus, a safe operating envelope for these components will be defined in the normal course of aircraft development. The presence of the autoland and RTA components do not alter this development activity. For this example, the Type I Safety Region can be defined to equal this envelope.

For the eVTOL example vehicle operating in hover mode, this envelope will place limits on a wide variety of vehicle states, including:

- Aerodynamic states: angle-of-attack (AOA), horizontal rate, vehicle accelerations, vertical rate, height-velocity state, advance ratio;
- Rotor states: rotor dynamic behavior including flap, vibrations, and elastic modes, rotor blade stall, as well as danger states such as Vortex Ring State (VRS) and wind milling;
- Actuator states: the actuators comprise a combination of electro-mechanical components that have both rate and physical limits; (There may also be provisions to not constantly operate an actuator near its physical limit to decrease maintenance costs and provide control margin for maneuverability.)
- Propulsion states: power available, power rates, and margins;
- Structural states: fuselage loads, actuator loads, and rotor loads / torques; (In addition to loads, the elastic modes and frequency excitation of the vehicle will typically be limited to ensure a resonance with the controller does not develop.)
- Spatial states: allowable distances and relative velocities between the aircraft and surrounding terrain, nearby aircraft, and other obstacles.

The Type I Safety Boundary is a complex hyper-surface over these state variables. It is typically developed through a series of analytical studies, simulation tests, ground testing of individual components, and test of systems in a build-up process. For some of these states, complying with upper and lower limits may keep the system in a safe state, while other states have more complicated constraint relationships. Structural states of UAM-type vehicles with multiple rotors, aerodynamic surfaces, and material construction featuring lighter weight but complex mechanical properties result in a complex interaction.

The Type I Safety Region will typically be expressed as a list of acceptable ranges for each state variable and of functional inequalities those variables must jointly satisfy. The Type I Safety Boundary separates those state combinations that satisfy each range and inequality in the list from those that fail to satisfy one or more of those constraints.

While it is beyond the scope of this report to detail the full construction of the Type I boundary, we take a brief look at an example involving the interaction of multiple simultaneous constraints. The scenario of interest is illustrated in Figure A-37. In this example, a landing aircraft must maintain a minimum separation distance from all vertical surfaces, such as a partial wall surrounding the landing site. At the same time, there are complicated constraints involving the aircraft's horizontal velocity and vertical descent rate to ensure that it does not enter a vortex ring state (VRS) or windmill brake state (WBS) [119][120]. The separation and velocity constraints could come into conflict if the aircraft were to approach a vertical surface at a high velocity and descent rate. For example, a large negative acceleration, which would halt the vehicle's forward motion, could cause the vehicle to enter a VRS.

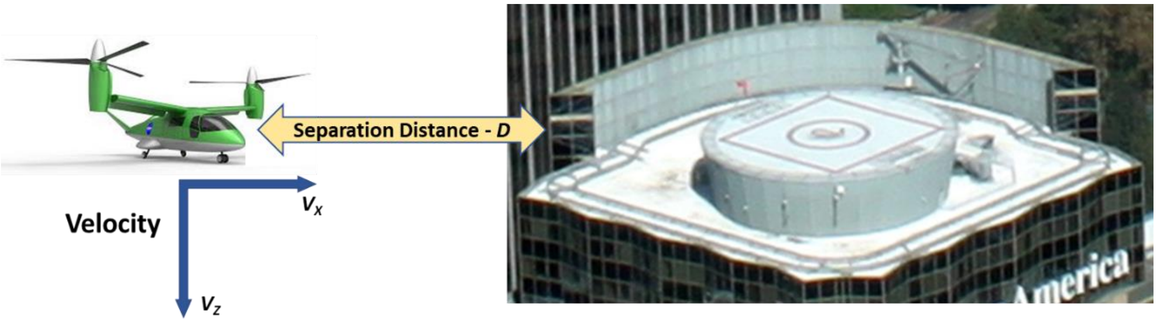


Figure A-37 RTA Scenario Involving Separation and Velocity

In [121], Johnson reviewed VRS theory and more than 50 experiments to derive a parametric VRS model suitable for real-time simulation and pilot training. The model includes empirical corrections to the momentum-theory model and was evaluated for single-shaft helicopters and tiltrotors. Due to the number of rotors on many UAM vehicle designs, and the reference model used in this study, the model developed by Johnson would need to be validated prior to actual usage but the theory and steps outlined in the report offer a foundation for use with UAM vehicles.

The vehicle operates in a VRS while descending at low advance ratio, or forward speed, and the descent rate approaches that of the induced wake velocity at the rotor disk. In this state, the rotor tip vortices are not pushed from the rotor disk quickly enough and are trapped around the rotor by the air mass coming from beneath the rotor. The tip vortices then collect in a ring, generating a re-circulatory flow through and around the rotor disk. If the forward speed of the vehicle is sufficiently fast the vortex ring is not developed.

Hence, a boundary for VRS development can be parameterized by the forward velocity V_x and vertical climb rate V_z (where climbing is positive). Both of these values are typically normalized by the induced velocity at hover, which from momentum theory is given by:

$$v_h = \sqrt{T^h/2\rho A_d} \qquad \text{Eq 13}$$

where T^h is the thrust required at hover, A_d is the area of the rotor disk, and ρ is air density.

Operation in the VRS introduces an unsteady condition, largely caused by asymmetrical and uneven forces and moments generated by the rotor. For helicopters, this results in an acute increase in descent rate and for tiltrotors a sharp roll-off. Despite the different manifestations for helicopters and tiltrotors, Johnson notes that the flight test data "…define essentially the same VRS boundary…This implies that basically the same aerodynamic mechanism is responsible for the behavior of both helicopter and tiltrotors in VRS" [121]. The VRS boundary is shown in Figure A-38 for helicopters, tiltrotor, and the VRS model Johnson developed.

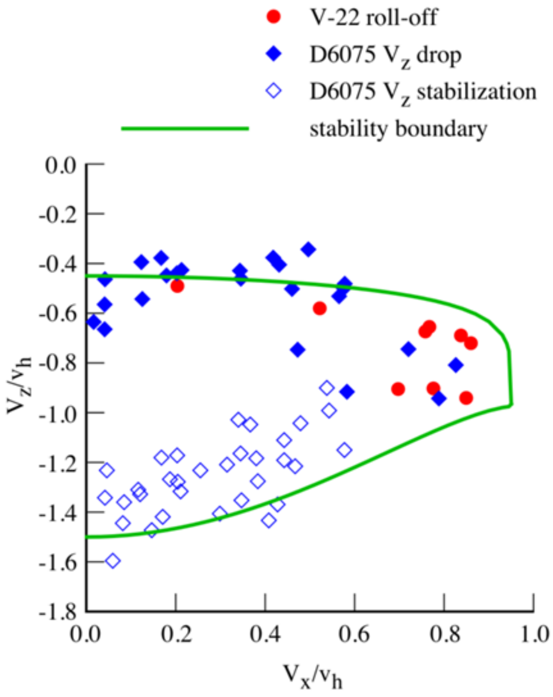


Figure A-38 VRS Boundary [121]

The Type I Safety Region will exclude the points inside the green boundary shown in Figure A-38. Typically, data to construct this boundary is gathered from a combination of flight tests and high-fidelity simulation. The conditions for the boundary can be based on flight dynamics, and / or uncomfortable / rough flight caused by mean fluctuations in the thrust. Following the method outlined by Johnson a dynamical model of the VRS can be developed such that it can be used for training UAM pilots and plan strategies to avoid the VRS region or escape from it.

For this example scenario, we model the VRS region based on an approximation of the eCRM-001 vehicle's induced velocity at hover. We form this approximation by first considering the thrust distribution amongst the 6 rotor systems. We make the following assumptions in this analysis:

1. Standard atmospheric conditions
2. Each rotor system (single or stacked) has the same diameter of 9 feet
3. The induced velocity value calculated from momentum theory applies to the stacked rotor
4. Ignore flat plate drag affects and interference between the rotor and fuselage
5. The vehicle center-of-gravity (CG) is aligned along the same station line as the mid- or wing-rotors, near the aft passenger row
6. The rotors are the sole source of aerodynamic force generation

The induced velocity can be calculated from momentum-theory as shown in Eq 13. The thrust for each rotor needs to be calculated for a hover trim condition; the remaining variables are known. Desired thrust for each rotor system is calculated for a hover condition where the forces and moments are balanced. The following conditions are met:

1. The total thrust of all rotor systems equals the gross weight, 5,130 lb.
2. The pitching moment generated by the wing-tip and aft rotors balance at a pitch attitude of zero degrees.
3. The roll moment generated by the wing-tip and mid-rotors on the left and right sides balance at a roll attitude of zero degrees.

The individual thrust values are calculated by solving a non-negative least-square problem of the form:

$$\min_{x} ||Ax - b||_2^2 \qquad \text{Eq 14}$$

where $A \in \Re^{4\times6}$, $x \in \Re^6$ is a vector of thrusts for each of the six rotors, and $b \in \Re^6$ is a vector defining desired conditions. Here the A matrix encodes the conditions required for a hover trim condition:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_{lf} & x_{rf} & 0 & 0 & x_{a1} & x_{a2} \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \qquad \text{Eq 15}$$

and $b = \{GW, 0,0,0\}^T$, where GW is the gross weight of the vehicle. The variables x_i in Eq 15 are the moment arms of the left front (lf), right front (rf), aft 1 (a1), and aft 2 (a2) rotors measured in the body-frame relative to the CG. The table below shows the least-squared error estimates of thrust for each rotor and the corresponding induced velocity at hover. In considering a VRS condition, the rotor the exhibits the smallest induced velocity should be used in the overall vehicle analysis as this is rotor that will experience VRS behavior at the lowest descent rate. For this example, this is the A2 rotor with a value of 37.24 ft/sec.

Table A-6 Individual Rotor Thrust and Induced Velocity Estimates

	LF	RF	LM	RM	A1	A2
Thrust [lbf]	1228.7	1228.7	854.96	854.96	542.61	420.02
Induced Velocity [ft/sec]	63.7	63.7	53.14	53.14	42.33	37.24

Developing a high-fidelity dynamic model of the eCRM-001 aircraft is beyond the scope of this project. As an alternative, we perform subsequent development based on a kinematic point-mass model of the aircraft, assuming that accelerations are limited to 1g (32.2 ft/sec^2) in the vertical direction and 2g (64.4 ft/sec^2) in the horizontal direction. While this simplified model is suitable for demonstrating a basic process for deriving RTA requirements, it ignores complicated dynamics associated with vehicle aerodynamics, non-instantaneous actuator response, internal dynamics of the TRC/ACAH/Mixing components, etc. In a practical application, these must all be accounted for.

Additionally, to simplify analysis and the design of a TRC controller for this vehicle, we approximate the VRS with an ellipsoid, as shown in Figure A-39. The ellipsoid is given by all pairs $[v_x, v_z]^T$ that satisfy the quadratic inequality

$$E\left(\begin{bmatrix} v_x \\ v_z \end{bmatrix}\right) \leq 1, \qquad \text{Eq 16}$$

where

$$E\left(\begin{bmatrix} v_x \\ v_z \end{bmatrix}\right) = \left(\begin{bmatrix} v_x \\ v_z \end{bmatrix} - \begin{bmatrix} 0 \\ -0.96 \end{bmatrix}\right)^T \begin{bmatrix} 1.1026 & -0.3483 \\ -0.3483 & 3.0165 \end{bmatrix} \left(\begin{bmatrix} v_x \\ v_z \end{bmatrix} - \begin{bmatrix} 0 \\ -0.96 \end{matrix} \qquad \text{Eq 17}$$

This corresponds to a counter-clockwise rotation by 10 degrees of an ellipse with semi-major axis 0.98 and semi-minor axis 0.57.

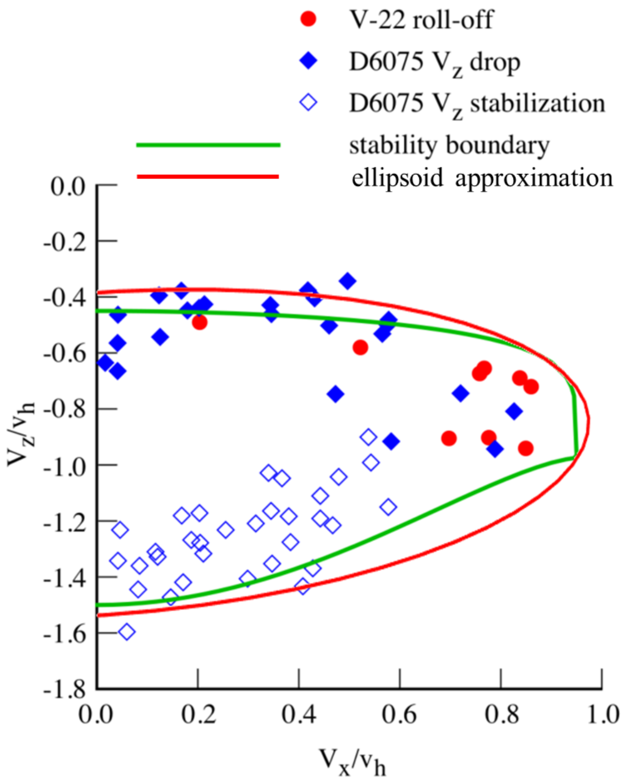


Figure A-39 VRS Ellipsoidal Approximation

As previously described, in this example the high-confidence path is designed to command a zero-velocity state. The example TRC will command a direct path through v_x, v_z space, if possible, as illustrated in the top-left panel of Figure A-40. In this illustration, the high-confidence path takes control when the vehicle has an initial velocity of 1.0 and -0.4 times v_h in the horizontal and vertical directions, respectively. It guides the vehicle directly to a zero velocity state on a straight-line path subject to the vehicle's acceleration limits. If a straight-line path to the top-left corner is not possible without entering a VRS, the high-confidence control path will choose a sequence of velocity commands that skirt the region, as shown in the top-right diagram of the figure. The TRC controller computes this path by finding, at every time step, a velocity vector of the form $v_c = (v_{xmin}, 0)$ such that a straight-line path to v_c will avoid the VRS and v_{xmin} is minimized.

The TRC has special logic to handle situations in which the vehicle is inside the VRS. A couple of different strategies for quickly exiting a VRS are taught to rotorcraft pilots. Details of these are overly complicated for this example analysis. As an alternative, we assume that the eCRM-001's TRC controller implements a strategy of commanding a maximum horizontal acceleration with a zero vertical acceleration, in an attempt to exit the VRS region through the right of the boundary. This is illustrated in the bottom diagram of Figure A-40.

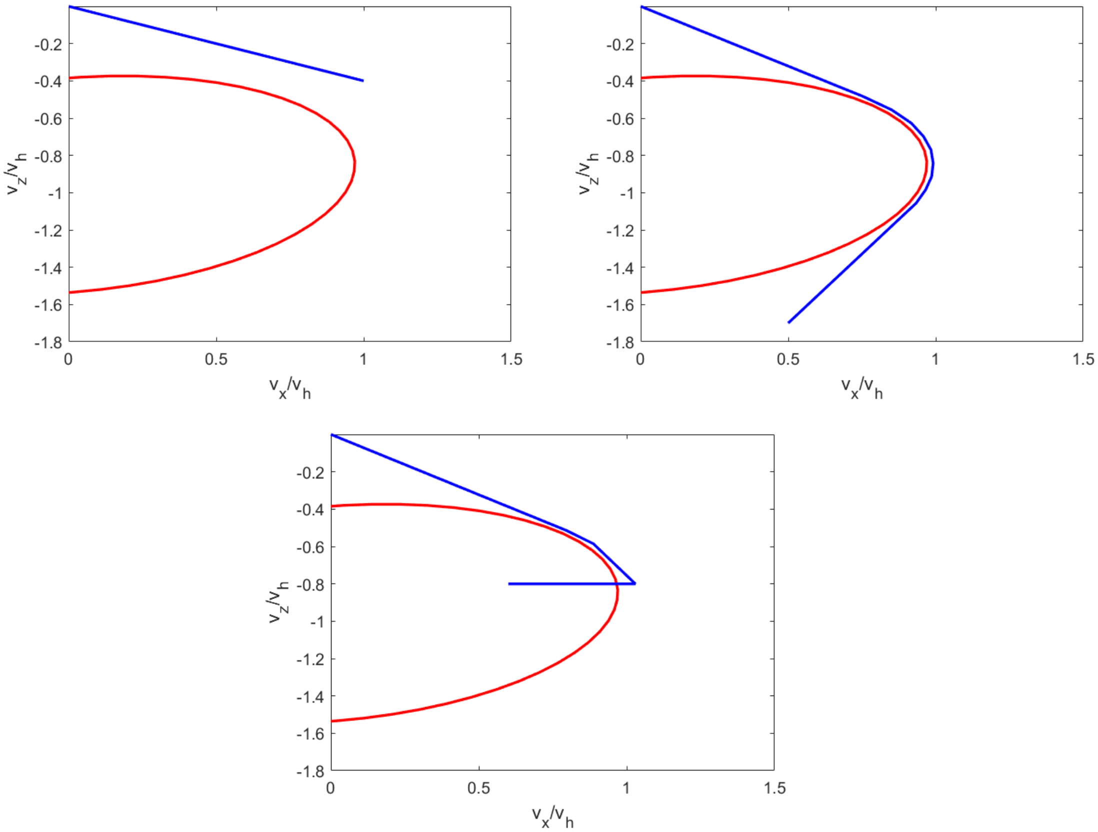


Figure A-40 Example Trajectories for the High-Confidence Control Path

This need to avoid VRS, or to exit it if necessary, means that the time required to bring the vehicle to a zero-velocity state is a complicated function of the initial velocity. The set of initial velocities from which the high-confidence control path can drive the aircraft to a zero-velocity state within the 2-second requirement is illustrated by the green region in Figure A-41. Figure A-42 shows the intersection of these velocity states with those that are outside the VRS. The velocity states in green denote the Type I Safety Region as projected onto the $V_x - V_z$ plane.

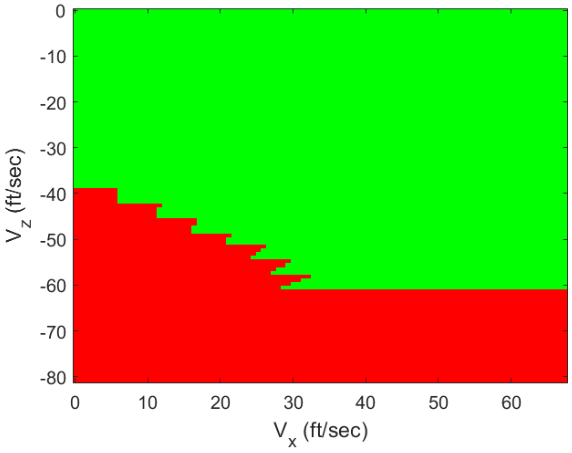


Figure A-41 States from which *Q* = [0,0] can be Reached within *T* = 2s

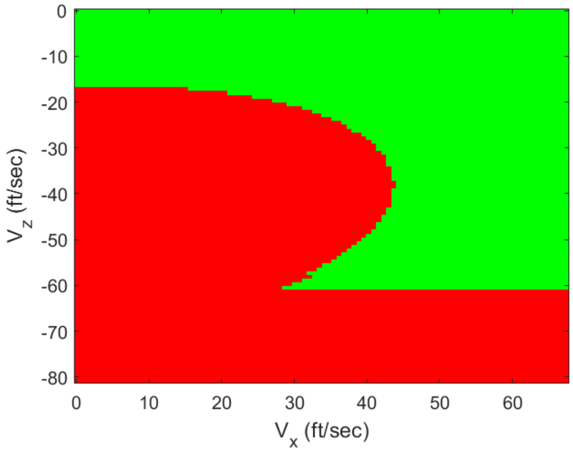


Figure A-42 Type I Safety Region Projected onto the $V_x - V_z$ Plane

As previously described, the Type I Safety Region involves separation distance as an additional dimension. We define the minimum safe separation distance from any vertical surface to be 100 feet. *Thus, the Type I Safety Region for this example consists of those vehicle states that have a separation distance of greater than 100 feet from any vertical surface and that have a velocity vector corresponding to the green region in Figure A-42.*

Type II Safety Boundary Determination

From an RTA safety perspective, the most important region of the state space is the exterior boundary of the Type II safety region. If the aircraft ever crosses this boundary, the RTA cannot guarantee that the reversionary controller will maintain flight safety. As a result, it is critically important to establish this boundary with high accuracy. However, for systems with even a moderate amount of complexity, a closed-form representation of that boundary may not be available through direct analysis. The focus is thus on constructing a sufficient approximation of this boundary.

The Type II Safety Boundary partitions the aircraft state space into two subsets:

1. States inside the Type II Safety Boundary – the set of states from which the high-confidence control path can assume control, then drive the aircraft to a state in *Q* within time *T*, all while keeping the vehicle state within the Type I Safety Region;
2. States outside the Type II Safety Region – the set of states from which the resulting state trajectory exits the Type I Safety Region.

More formally, suppose that the aircraft was in state s when the RTA components switched from the low-confidence control path to the high-confidence control path. Let the function $R(s, t)$ denote the resulting aircraft state *t* seconds after this switch; $R(s, t)$ is the state trajectory originating at state s. If $R(s, t)$ stays within the Type I Safety Region for all values of t, then the state s is within the Type II Safety Region. Otherwise, s is outside the Type II Safety Region.

Our interest is with those initial states s that lie on the boundary between these sets. To formalize this notion, we define the function $D(s)$ that, for an arbitrary state s, equals the distance from s to the boundary of the Type I Safety Region, the construction of which was discussed in the previous section. We define $D(s)$ to be a signed distance function, meaning that its values are positive when s is inside the Type I Safety Region, and they are negative when s is outside that region. That is, $D(s)$ will have values that are close to zero for states that are near the Type I Safety Boundary, large positive values for states that are well inside the boundary, and large negative values for states that are far outside the boundary. Note that there is some flexibility in choosing the kind of distance measure to use. For example, the magnitude of $D(s)$ might equal the Euclidian distance between s and the closest point on the Type I Safety Boundary, with a sign that is positive or negative depending on which side of the boundary it is on.

Finally, we define the function $g(s)$ which equals:

$$g(s) = \min_{t \in (0,T)} D(R(s,t)) \qquad \text{Eq 18}$$

The significance of $g(s)$ is as follows: If $g(s) > 0$, then the RTA components can switch to the high-confidence control path when the aircraft is in state s, and the resulting state trajectory will stay entirely within the Type I Safety Region. On the other hand, if $g(s) < 0$, then switching to the high-confidence control path may cause the aircraft to exit the Type I Safety Region. The Type II Safety Boundary is the set of states such that $g(s) = 0$.

In practice, we need the ability to compute the function $g(s)$ from simulation or flight-test; we do not necessarily require a closed-form representation of the function.

For the eCRM-001 example, relevant vehicle states constitute a triple $s = [v_x, v_z, d]^T$, where v_x is the aircraft's forward velocity, v_z is the aircraft's vertical velocity, and d is the separation distance to the vertical surface. The function $R(\cdot,\cdot)$ is implemented in terms of the kinetic model previously discussed, and the distance function $D(\cdot)$ is computed as

$$D(s) = \min(E(s) - 1, 2 - T(s), d - 100), \qquad \text{Eq 19}$$

where $E(s)$ defines the VRS ellipse (defined in Eq 17), $T(s)$ is the time required by the high-confidence controller to achieve a zero-velocity state, and d is defined to be negative if the aircraft passes across the boundary of the vertical surface.

Time Horizon Determination

The time horizon parameter τ is chosen to be an upper bound on how much time might elapse between the generation of a faulty command by the low-confidence control path and a subsequent switch to the high-confidence control path. This quantity must take into account the rate at which the RTA monitor performs command checking and the amount of time required for the RTA switching mechanism to begin transmitting commands from the high-confidence path.

For the eCRM-001 vehicle, we assume that the RTA monitor operates at a rate at least 10Hz, and that the switching time is negligible. We thus choose $\tau = 0.1$s for this example.

Figure A-43 illustrates the significance of the quantity, τ. In this failure scenario example, the vehicle is initially in a state that is within the Type II Safety Region. The autoland algorithm erroneously commands a max-acceleration maneuver, forward and down for 1/10 sec (the green line segment), before the RTA monitor detects the problem and switches to the high-confidence path for recovery (the blue line segments). In that brief time frame, the vehicle has entered VRS. The Type III Safety Region is defined relative to τ to prevent this kind of situation.

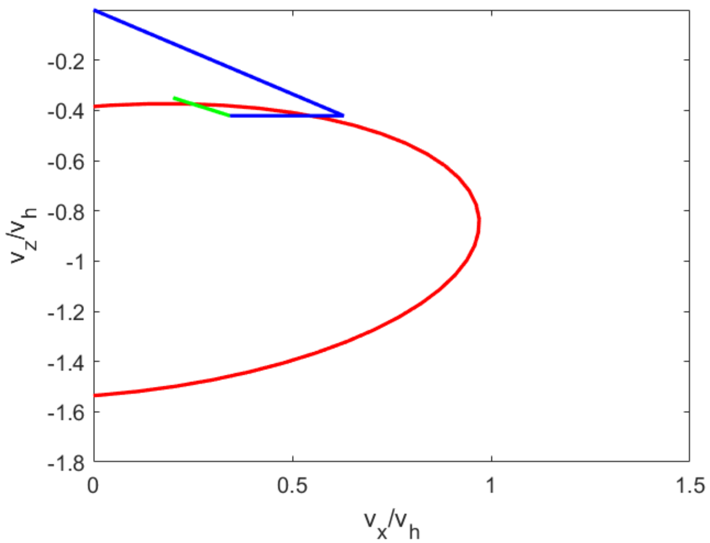


Figure A-43 Failure Scenario that must be Prevented by RTA

Type III Safety Boundary Determination

Similar to the Type II case, we want to define a function $h(s)$ with the property that $h(s) > 0$ for states that are inside the Type III Safety Region, $h(s) < 0$ for states outside the Type III Safety Region, and $h(s) = 0$ for states that are on the Type III Safety Boundary. With an ability to compute this function, the RTA monitor can simply evaluate h at the current state, and switch to the high-confidence control path if that evaluation is ever less than or equal to zero.

To define this function, $h(s)$, recall that, for any point outside the Type III Safety Region, the low-confidence controller could drive the aircraft outside the Type II Safety Region within τ seconds. The Type III Safety Region is thus the set of points that are far enough inside the Type II Safety Boundary that the low-confidence controller is not capable of exiting the Type II Safety Region in that amount of time. (Recall that, since we cannot know exactly what the advanced controller *will* do in any given situation, we define the switching boundary in terms of what the advanced controller *can* and *cannot* do). With this in mind, a state s based on the trajectory that would result from the worst-case control command applied for τ seconds, followed by the state trajectory $R(s,t)$ that would result from switching to the high-confidence path, can be evaluated. If this entire trajectory stays within the Type I Safety Region, then s is within the Type III safety Region.

Stated more formally, let the aircraft's state equations be defined by the differential equation

$$\dot{s} = f(s,u), \qquad \text{Eq 20}$$

where u is an arbitrary command that could be generated by the low-confidence control path. Let the solution to this differential equation after τ seconds be denoted $\sigma(s,u,\tau)$. Then the function Eq 21 is the worst-case signed distance to the Type I Safety Boundary after switching to the high-confidence controller that could result if RTA monitor waited τ seconds before commanding such a switch. Thus, this definition of h has exactly the properties we require.

$$h(s) = \min_{u} g(\sigma(s,u,\tau)) \qquad \text{Eq 21}$$

Note that Eq. 21 is an optimal control problem, which in general will not have a closed form solution. In practice, Eq 21 can be computed for an arbitrary state s from a simulation model of the system, by numerically searching for the worst-case control command sequence. When τ is small relative to the time scale of aircraft dynamics, the process can be highly simplified by appropriate linearization of $f(s,u)$ and $g(s)$.

Finally, we define a practical representation of the function $h(s)$ and a mechanism for approximating it in the context of a specific vehicle and high-confidence control path. We note that for physically realizable aircraft, $h(s)$ will be defined over a finite domain and will have finite energy. Thus, it can be approximated to an arbitrary accuracy as a linear combination of basis functions, with the form:

$$h(s) \approx \sum_{n=0}^{N} a_n \phi_n(s), \qquad \text{Eq 20}$$

where the family of functions $\{\phi_n(s)\}$ form a complete orthonormal set of functions over the aircraft's state space. The theory of orthogonal series representations is well-developed, and many such function families have been studied in depth. Selection of a specific family for this application is somewhat arbitrary and can be based on engineering judgment, computational expediency, etc. Common examples of the functions $\phi_n(s)$ in related applications include complex exponential functions, in which case the approximation becomes a multivariate Fourier series, and Legendre polynomials, in which case the approximation becomes a multivariate polynomial. An arbitrary degree of accuracy can be obtained by increasing the series order, N.

Once the function family $\{\phi_n(s)\}_{n=0}^{N}$ has been chosen, the coefficients $\{a_n\}_{n=0}^{N}$ can be computed from simulation data of the high-confidence control path and a model of the aircraft dynamics. Let the data from M simulation runs be given as $\{(s_m, h(s_m))\}_{m=1}^{M}$, where s_m is an initial state, and $h(s_m)$ is found by the simulation-based methods described in the previous section. From this data, we can choose the coefficients $\{a_n\}_{n=0}^{N}$ to minimize the sum of squared differences between these computed $h(s_m)$ values and the approximation in Eq 20:

$$\min_{a_0,a_1,\ldots,a_N} \sum_{m=1}^{M} [h(s_m) - \sum_{n=0}^{N} a_n \phi_n(s)]^2. \qquad \text{Eq 21}$$

Note that the exact value of $h(s)$ is not as important to the application as its sign, which indicates whether or not the state s is in the Type III Safety Region. To ensure that the resulting approximation has the correct sign for each simulated state, we can solve the above optimization problem subject to the linear constraints in Eq 22.

$$h(s_m) \sum_{n=0}^{N} a_n \phi_n(s_m) \geq 0, \qquad \text{Eq 22}$$

for all $m = 1, 2, \ldots, M$.

As with all model development activity of this type, the simulation initial conditions, $\{s_m\}_{m=0}^{M}$ should be chosen based on experienced engineering judgment, with a special focus on states that are near the Type II Safety Boundary. This process is typically iterative, with a gradual increase in sample data and model order N until there is very good agreement between the simulation results $h(s_m)$ and the approximation in Eq. 20.

To demonstrate this process, an approximation of $h(s)$ in the form of Eq 21 was constructed using the uncertainty quantification tool **AURA** (a tool developed and distributed by Barron Associates, Inc.). The basis functions ϕ_n were chosen to be the multi-dimensional Legendre polynomials. The number of terms N and number of simulations M were chosen to achieve a mean-squared error of less than 1 part in 10,000 between $h(s)$ and its series approximation. Samples of $h(s)$ were constructed by simulating the worst-case command that an autoland controller could generate, and then allowing that command to be active for one-tenth of a second, which is the time horizon τ. Following that brief period, the simulation switched to the high-confidence control design, which drove the vehicle to a zero-velocity state using the TRC controller previously discussed. The function $D(s)$ in Eq 19 was evaluated for each point in the resulting state trajectory, and its minimum value over the entire trajectory sequence was taken as the true value of $h(s)$.

In general, the set of states for which $h(s) = 0$ is a multi-dimensional hypersurface, which can be difficult to depict. In this example, however, it forms a two-dimensional surface within the three-dimensional state space $v_x \times v_z \times d$. This hypersurface can be visualized by showing the region of the V_x – V_z plane that is inside the Type III Safety Region for different values of d. This is illustrated by the green regions in Figure A-44, for *d* = 115ft, 130ft, 145ft, and 160ft.

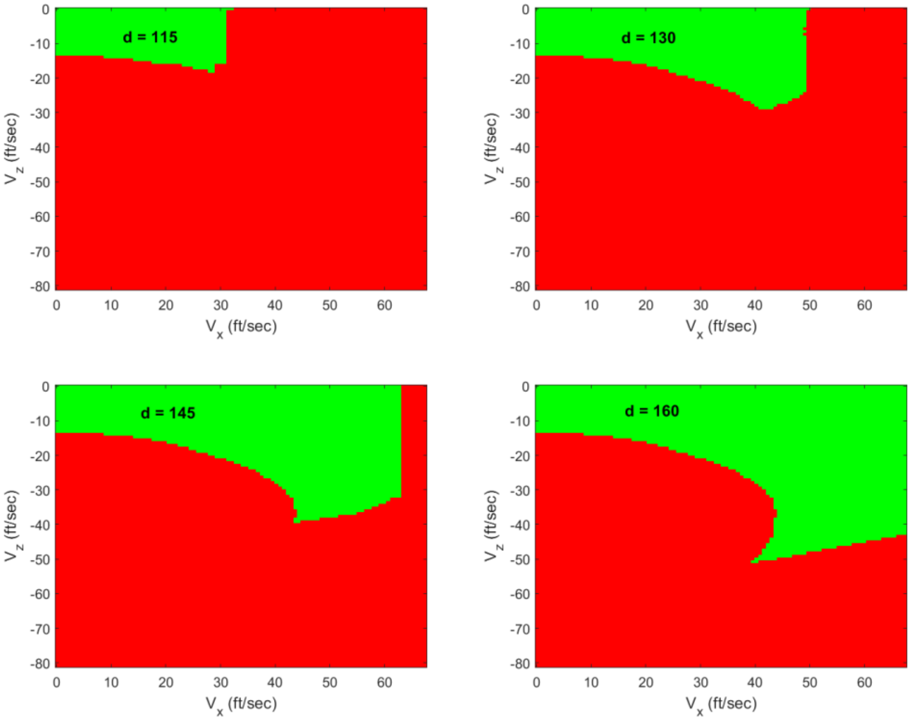
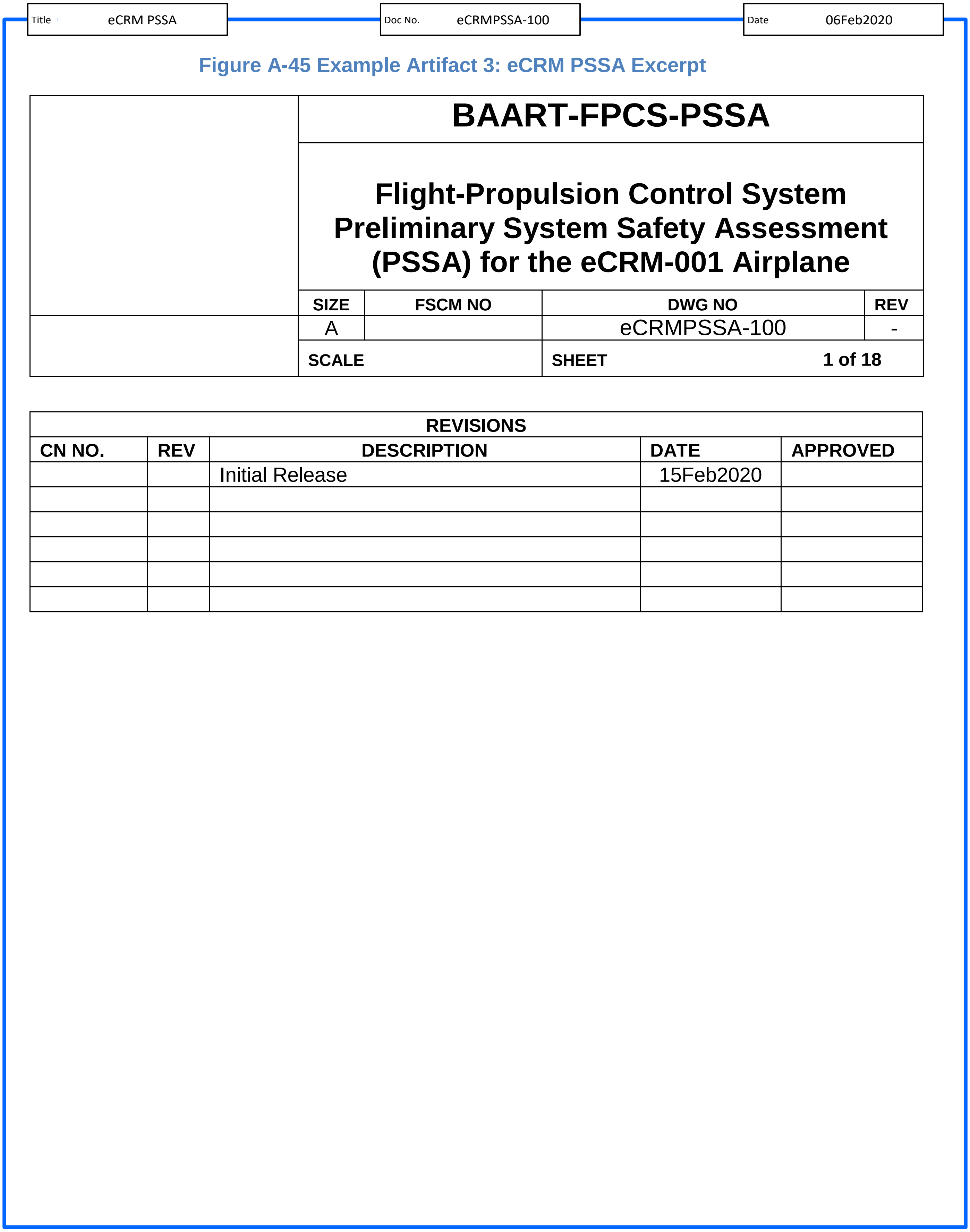


Figure A-44 Type III Safety Region in the V_x-V_z Plane for Different Values of *d*

A3.6 FPCS PSSA – Baseline Process

To support system design and develop necessary safety focused requirements, a preliminary system safety assessment (PSSA) is accomplished. The initial artifact associated with this effort is presented in Figure A-45 Example Artifact 3: eCRMPSSA-100 document.

Figure A-45 Example Artifact 3: eCRM PSSA Excerpt



Title	eCRM PSSA	Doc No.	eCRMPSSA-100	Date	06Feb2020

BAART-FPCS-PSSA

Flight-Propulsion Control System Preliminary System Safety Assessment (PSSA) for the eCRM-001 Airplane

SIZE	FSCM NO	DWG NO	REV
A		eCRMPSSA-100	-
SCALE		SHEET	1 of 18

REVISIONS				
CN NO.	REV	DESCRIPTION	DATE	APPROVED
		Initial Release	15Feb2020	

1. Introduction

The assessment captured herein represents the example Preliminary System Safety Assessment (PSSA) for the eCRM-001 Flight-Propulsion Control System (FPCS).

1.1. References

The following documents are referenced herein.

[1] 14CFR/CS Part 23, Amendment 23-64

[2] eCRM-001 Flight-Propulsion Control System Requirements Document

[3] ARP4761 "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Aircraft Systems and Equipment"

[4] ASTM F3230-17, "Safety Assessment of Systems and Equipment in Small Aircraft"

[5] eCRMSFHA-100, FPCS Functional Hazard Assessment for the eCRM-001 Airplane

[6] eCRMCONOP, eCRM-001 Concept of Operations Analysis *(Editor's Note: not developed in this example)*

Editor's Note: Document reference numbering within an example artifact will be to the documents listed as references in this section rather than the overall report reference list.

1.2. Glossary

This section captures specific terms and definitions used within the PSSA.

Term	Definition
Uncommanded	Activation of a function without pilot command input or erroneously activated due to equipment failure.
Minimum Acceptable Control (MAC)	An airplane configuration under which the normal acceptable control performance criteria will still be satisfied and when lost will result in the failure condition effects described.

2. System Description Summary

Editor's note: Duplicate system description summary removed for brevity.

3. Flight-Propulsion System PSSA Development

3.1. PSSA Inputs

A review of the Flight-Propulsion Requirements Document [2] identified the system level functions and implementation planned for the eCRM-001 FPCS. The captured required implementation has been used to formulate the PSSA failure models.

The eCRM-001 concept of operations analysis [6] has established the initial operation characteristics for the vehicle. This information has been repeated in Table PSSA-1 for reader convenience.

Table PSSA-1 eCRM-001 Operating Characteristics

Parameter	Characteristic
Average Flight Duration	0.5 hr (30 min)
Power-on Operating Time (I.E. time between electronics power-on)	3.0 hours
Airplane life	10 years

Note: Battery recharge time is not considered in operating time.

3.2. SFHA Failure Conditions

The following failure conditions and classifications from the FPCS SFHA are provided as inputs to the FPCS PSSA process.

Table PSSA-2 SFHA Failure Conditions and Classifications

FC ID number	Failure Condition	Flight Phase	Classification
1.1.1.TL	Loss of wingtip rotor lift	T1, T2, T3, L1, L2	Major
1.1.1.PL	Partial loss of wingtip rotor lift	T1, T2, T3, L1, L2	Minor
1.1.1.MF1	Uncommanded wingtip rotor lift – single rotor	T1, T2, T3, L1, L2	Major
1.1.1.MF2	Uncommanded wingtip rotor lift – dual rotor	T1, T2, T3, L1, L2	Hazardous
1.1.1.MF3	Erroneous wingtip rotor lift intensity – excessive	T1, T2, T3, L1, L2	Hazardous
1.1.1.MF4	Erroneous wingtip rotor intensity - diminished	T1, T2, T3, L1, L2	Hazardous
1.2.1.TL	Loss of wing lifting rotor stack lift	T1, T2, T3, L1, L2	Major
1.2.1.PL	Partial loss of wing lifting rotor stack lift	T1, T2, T3, L1, L2	Minor

Table PSSA-2 SFHA Failure Conditions and Classifications

FC ID number	Failure Condition	Flight Phase	Classification
1.2.1.MF1	Uncommanded wing lifting rotor stack lift – single rotor	T1, T2, T3, L1, L2	Major
1.2.1.MF2	Uncommanded wing lifting rotor stack lift – dual rotor	T1, T2, T3, L1, L2	Major
1.2.1.MF3	Erroneous wing lifting rotor stack lift intensity – excessive	T1, T2, T3, L1, L2	Minor
1.2.1.MF4	Erroneous wing lifting rotor stack lift intensity - diminished	T1, T2, T3, L1, L2	Minor
1.3.1.TL	Loss of fuselage rotor stack lift	T1, T2, T3, L1, L2	Major
1.3.1.PL	Partial loss of fuselage rotor stack lift	T1, T2, T3, L1, L2	Minor
1.3.1.MF1	Uncommanded fuselage rotor stack lift – single rotor	T1, T2, T3, L1, L2	Minor
1.3.1.MF2	Uncommanded fuselage rotor stack lift – dual rotor	T1, T2, T3, L1, L2	Major
1.3.1.MF3	Erroneous fuselage rotor stack lift intensity – excessive	T1, T2, T3, L1, L2	Minor
1.3.1.MF4	Erroneous fuselage rotor stack lift intensity - diminished	T1, T2, T3, L1, L2	Minor

Editors' Note: Only the "Control Lift for VTOL" system function is developed in this example. All other system functions, at the level of the functional breakdown defined in A4.1, would be developed in a similar fashion.

3.3. Assess Failure Condition Evaluation

The PSSA evaluates the preliminary system implementation architecture against the qualitative and quantitative evaluations established in the SFHA [5]. This architecture (including system description, interfaces and system requirements) is analyzed to determine if it can reasonably be expected to meet the safety objectives.

3.3.1. Exposure Times

Table PSSA-3 presents the exposure time values used in the fault tree evaluation of the failure conditions.

Table PSSA-3 FTA Exposure Times

Exposure Time Description	Time (Hrs)
Average flight time	0.5
Average time between automated self-test execution (airplane power up time)	3.0
Takeoff (T1, T2, T3)	0.0217 (78 sec)
Land (L1, L2)	0.035 (126 sec)

3.3.2. Basic Event Failure Rate Data

The general LRU reliability prediction values used in this analysis are shown in Table PSSA-3. Attempts have been made to limit the analysis to the use of LRU-level failure rates. However, there are instances where more accurate modeling is required to either accurately represent the system architecture or allow proper analysis in support of specific requirements. For these cases, the FCC failure probabilities are allocated to sub-LRU elements.

Table PSSA-3 FPCS LRU Reliability Predictions

Reference	LRU Description	MTBF (Flight Hours)	Failure Probability
TBD	Flight Control Computer (FCC)	65,000	1.54E-05
TBD	Wingtip Motor Controller	40,000	2.50E-05
TBD	Lift Motor Controller	40,000	2.50E-05
TBD	Surface EMA Controller	40,000	2.50E-05
TBD	HSTAB EMA Controller	25,000	4.00E-05
TBD	Wingtip Motor	100,000	1.00E-05
TBD	Fuselage Motor	100,000	1.00E-05
TBD	Prop. Pitch EM Actuator	80,000	1.25E-05
TBD	Nacelle Tilt EM Actuator	80,000	1.25E-05
TBD	Flow Control EM Actuator	80,000	1.25E-05
TBD	Surface Control EM Actuator	80,000	1.25E-05
TBD	HSTAB EM Actuator	40,000	2.50E-05
TBD	Wingtip Mechanical Gearbox	1,000,000	1.00E-06

Editor's Note: MTBF values based on engineering Rough Order of Magnitude (ROM) estimates

The following external events and their rates are assumed:

- Loss of electrical power provided by lift busses: 1.00 E-04 per flight hour
- Loss of electrical power provided by lift-cruise busses: 1.00 E-04 per flight hour.

3.3.3. Fault Tree Analysis (FTA)

Table PSSA-4 summarizes the results of the fault tree analysis for those failure conditions classified as catastrophic or hazardous. The FTA results indicate the planned system architecture will satisfy the SFHA safety objectives.

Table PSSA-4 FTA Results

FC Reference	FC Description	Flight Phase	Safety Objective (PFH/PFP)	Analysis Results
1.1.1.MF2	Uncommanded wingtip rotor lift – dual rotor	T1, T2, T3, L1, L2	AW II ≤ 1.0E-06	1.139E-09 per flight
1.1.1.MF3	Erroneous wingtip rotor lift intensity – excessive	T1, T2, T3, L1, L2		
1.1.1.MF4	Erroneous wingtip rotor intensity - diminished	T1, T2, T3, L1, L2		

Editor's Note: The FTA results for the selected failure conditions were based on average flight duration resulting in a per flight probability.

4. Derived System Requirements

Table PSSA-5 captures the proposed requirements necessary for the implementation to satisfy the safety objectives analyzed.

Table PSSA-5 Derived Safety Requirements

Requirement Identifier	Proposed System Safety Requirement
PSSA-1	Each FCC shall provide a Command (CMD) and Monitor (MON) Lane architecture. Rationale: Comparison monitoring needed to ensure high integrity calculation and transmission are achieved.
PSSA-2	The Command Lane and Monitor Lane of each FCC shall be independent. Rationale: Needed to mitigate common mode failure mechanisms of the comparison monitoring mechanism.

Table PSSA-5 Derived Safety Requirements

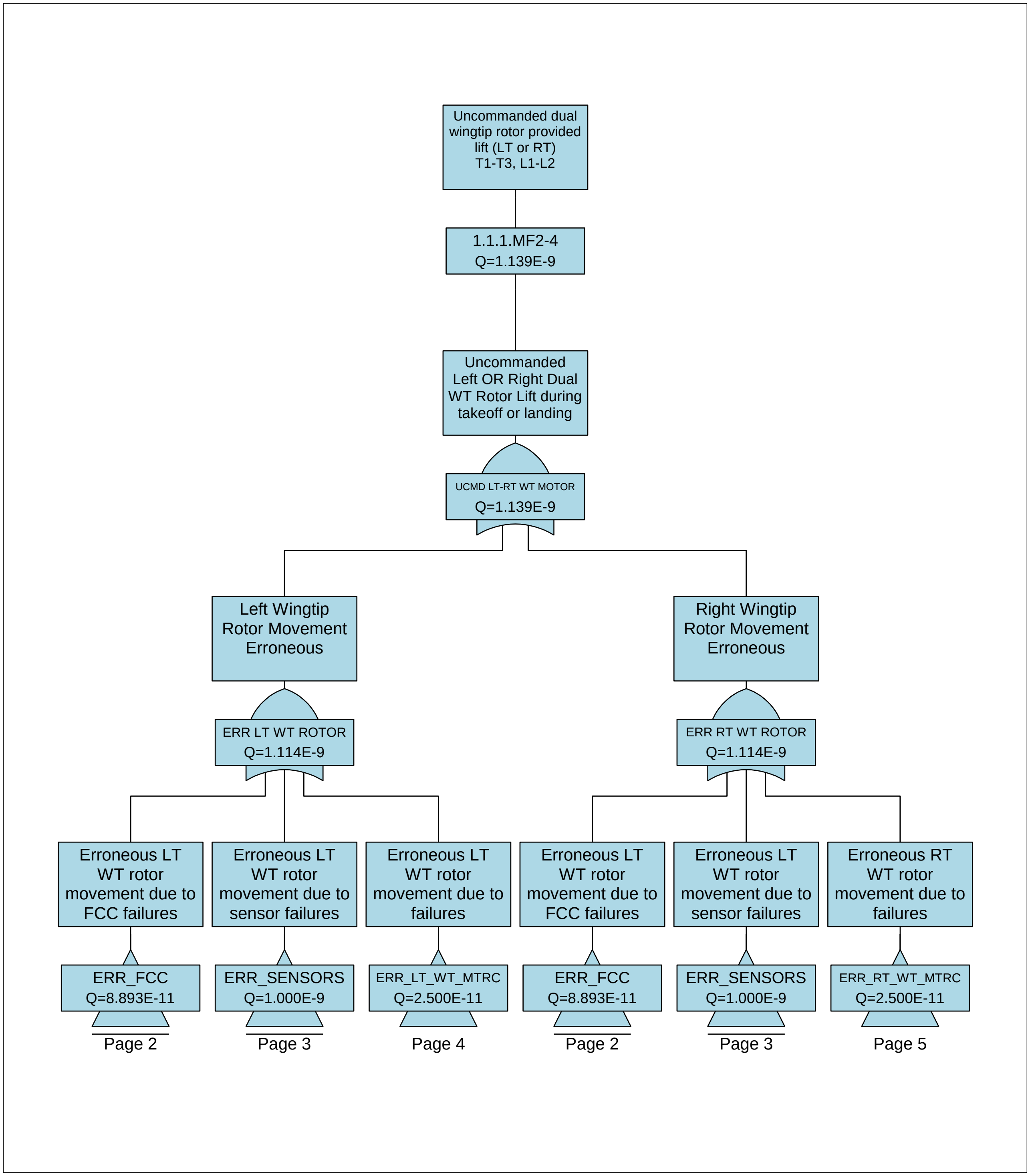
Requirement Identifier	Proposed System Safety Requirement
PSSA-3	Each Wing Tip Motor Controller shall each provide a Command (CMD) and Monitor (MON) Lane architecture. Rationale: Comparison monitoring needed to ensure a high integrity calculations are achieved for detection of motor and control electronics failures.
PSSA-4	The Command Lane and Monitor Lane of each Wing Tip Motor Controller shall be independent. Rationale: Needed to mitigate common mode failure mechanisms of the comparison monitoring mechanism.
PSSA-5	Each Lift Motor Controller shall each provide a Command (CMD) and Monitor (MON) Lane architecture. Rationale: Comparison monitoring needed to ensure a high integrity calculations are achieved for detection of motor and control electronics failures.
PSSA-6	The Command Lane and Monitor Lane of each Lift Motor Controller shall be independent. Rationale: Needed to mitigate common mode failure mechanisms of the comparison monitoring mechanism.

5. FTA Analysis Discussion

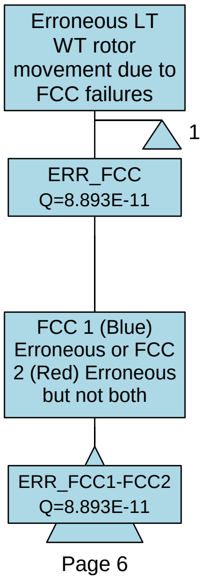
Editor's Note: A PSSA would normally contain additional qualitative evaluations in addition to the quantitative evaluation presented in Table PSSA-4. These characteristics have been omitted for example brevity.

PSSA Appendix A FTA Listing

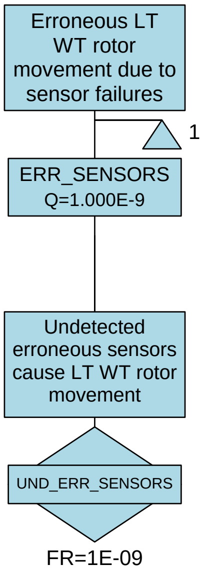
eCRM-001
FTA


Uncommanded dual wingtip rotor provided lift (LT or RT) T1-T3, L1-L2
1.1.1.MF2-4
Q=1.139E-9
Uncommanded Left OR Right Dual WT Rotor Lift during takeoff or landing
UCMD LT-RT WT MOTOR
Q=1.139E-9
Left Wingtip Rotor Movement Erroneous
ERR LT WT ROTOR
Q=1.114E-9
Right Wingtip Rotor Movement Erroneous
ERR RT WT ROTOR
Q=1.114E-9
Erroneous LT WT rotor movement due to FCC failures
ERR_FCC
Q=8.893E-11
Page 2
Erroneous LT WT rotor movement due to sensor failures
ERR_SENSORS
Q=1.000E-9
Page 3
Erroneous LT WT rotor movement due to failures
ERR_LT_WT_MTRC
Q=2.500E-11
Page 4
Erroneous LT WT rotor movement due to FCC failures
ERR_FCC
Q=8.893E-11
Page 2
Erroneous LT WT rotor movement due to sensor failures
ERR_SENSORS
Q=1.000E-9
Page 3
Erroneous RT WT rotor movement due to failures
ERR_RT_WT_MTRC
Q=2.500E-11
Page 5

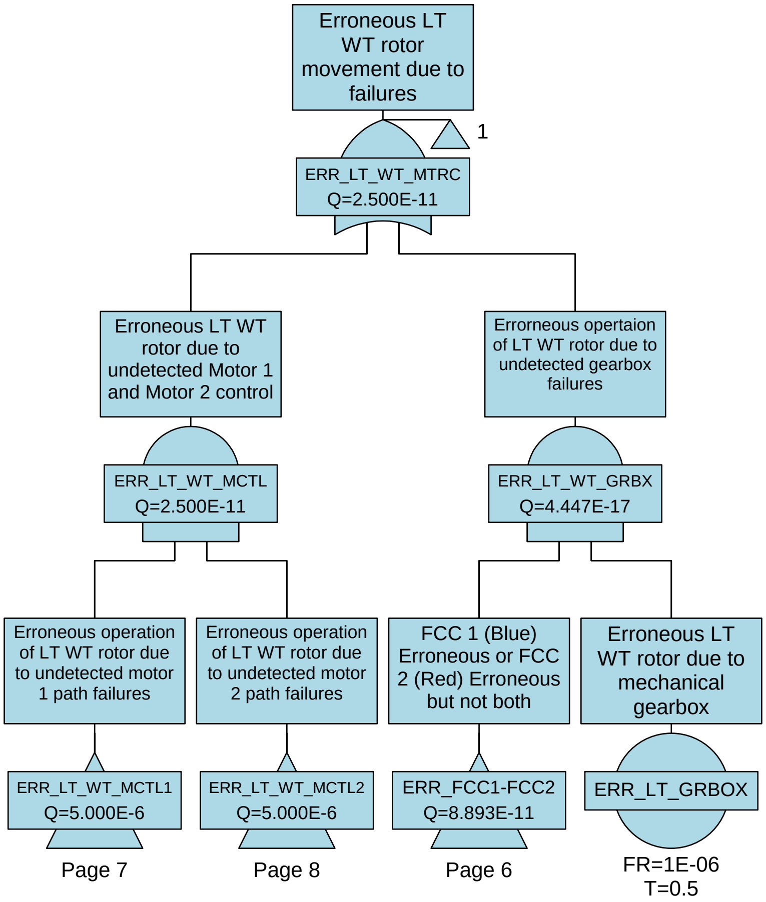
eCRM-001
FTA


Erroneous LT WT rotor movement due to FCC failures
1
ERR_FCC
Q=8.893E-11
FCC 1 (Blue) Erroneous or FCC 2 (Red) Erroneous but not both
ERR_FCC1-FCC2
Q=8.893E-11
Page 6

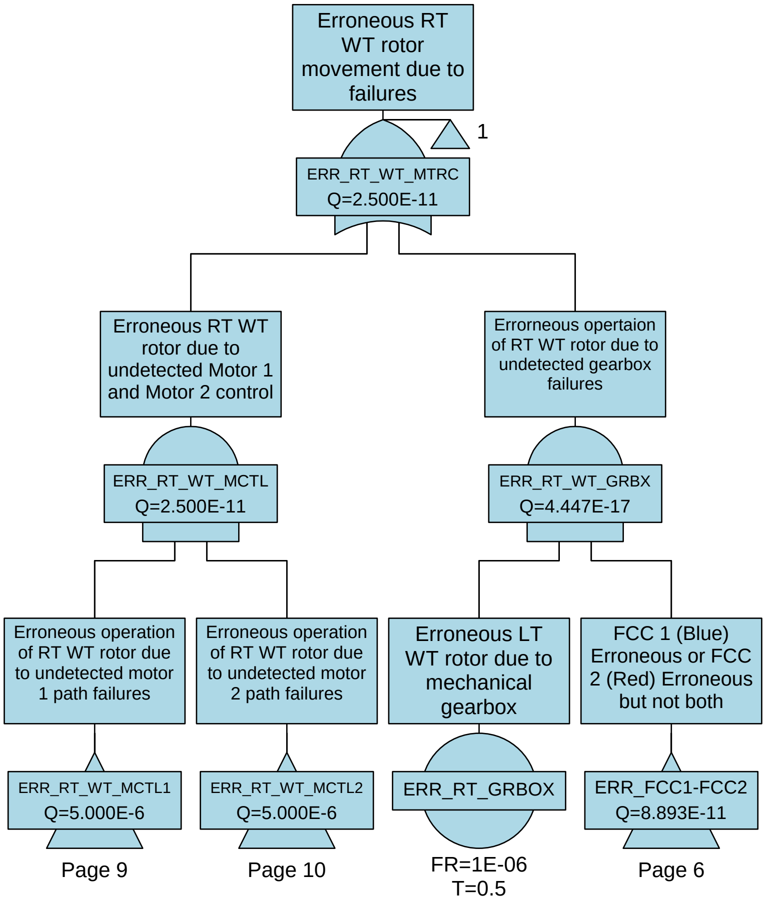
eCRM-001
FTA


Erroneous LT WT rotor movement due to sensor failures
1
ERR_SENSORS
Q=1.000E-9
Undetected erroneous sensors cause LT WT rotor movement
UND_ERR_SENSORS
FR=1E-09

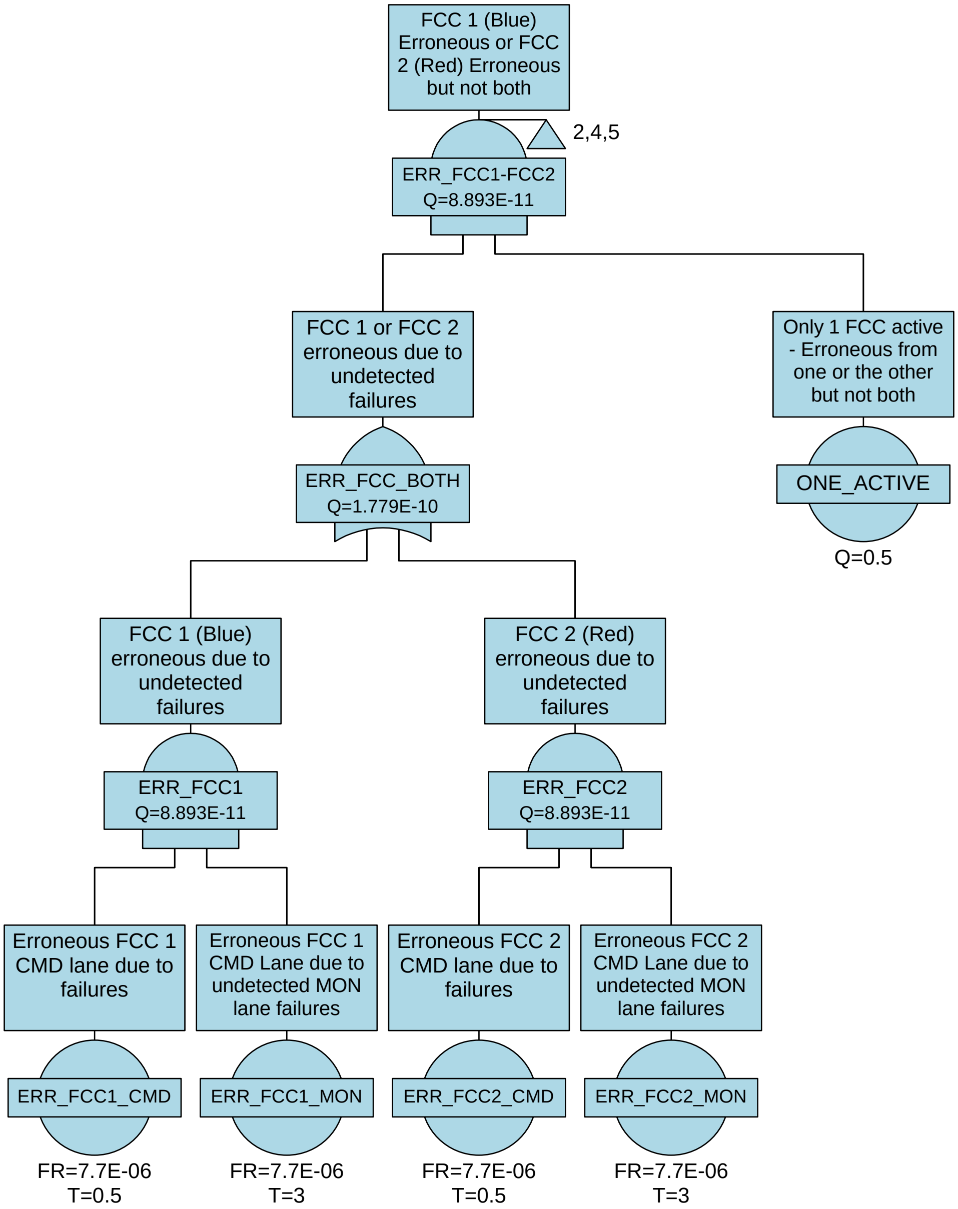
eCRM-001
FTA


Erroneous LT WT rotor movement due to failures
1
ERR_LT_WT_MTRC
Q=2.500E-11
Erroneous LT WT rotor due to undetected Motor 1 and Motor 2 control
ERR_LT_WT_MCTL
Q=2.500E-11
Errorneous opertaion of LT WT rotor due to undetected gearbox failures
ERR_LT_WT_GRBX
Q=4.447E-17
Erroneous operation of LT WT rotor due to undetected motor 1 path failures
ERR_LT_WT_MCTL1
Q=5.000E-6
Page 7
Erroneous operation of LT WT rotor due to undetected motor 2 path failures
ERR_LT_WT_MCTL2
Q=5.000E-6
Page 8
FCC 1 (Blue) Erroneous or FCC 2 (Red) Erroneous but not both
ERR_FCC1-FCC2
Q=8.893E-11
Page 6
Erroneous LT WT rotor due to mechanical gearbox
ERR_LT_GRBOX
FR=1E-06
T=0.5

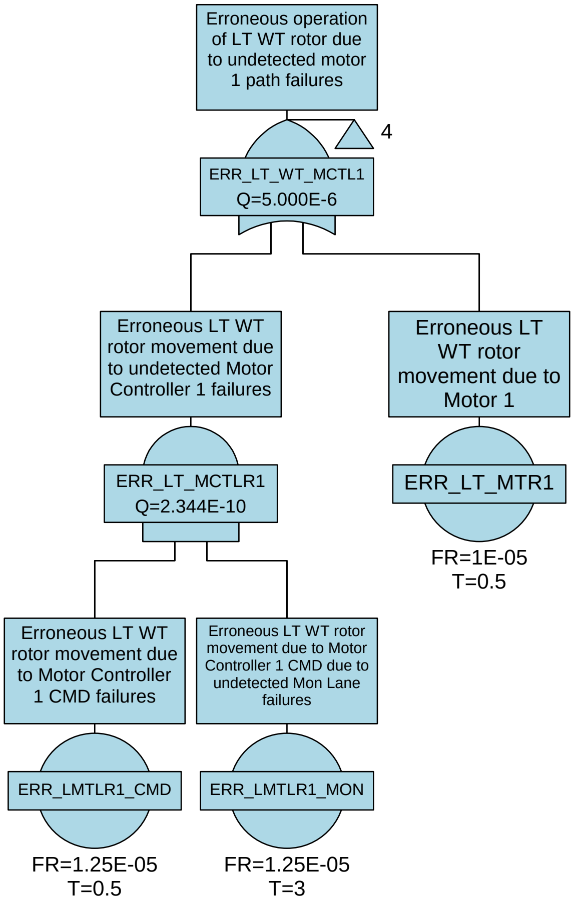
eCRM-001
FTA


Erroneous RT WT rotor movement due to failures
1
ERR_RT_WT_MTRC
Q=2.500E-11
Erroneous RT WT rotor due to undetected Motor 1 and Motor 2 control
ERR_RT_WT_MCTL
Q=2.500E-11
Errorneous opertaion of RT WT rotor due to undetected gearbox failures
ERR_RT_WT_GRBX
Q=4.447E-17
Erroneous operation of RT WT rotor due to undetected motor 1 path failures
ERR_RT_WT_MCTL1
Q=5.000E-6
Page 9
Erroneous operation of RT WT rotor due to undetected motor 2 path failures
ERR_RT_WT_MCTL2
Q=5.000E-6
Page 10
Erroneous LT WT rotor due to mechanical gearbox
ERR_RT_GRBOX
FR=1E-06
T=0.5
FCC 1 (Blue) Erroneous or FCC 2 (Red) Erroneous but not both
ERR_FCC1-FCC2
Q=8.893E-11
Page 6

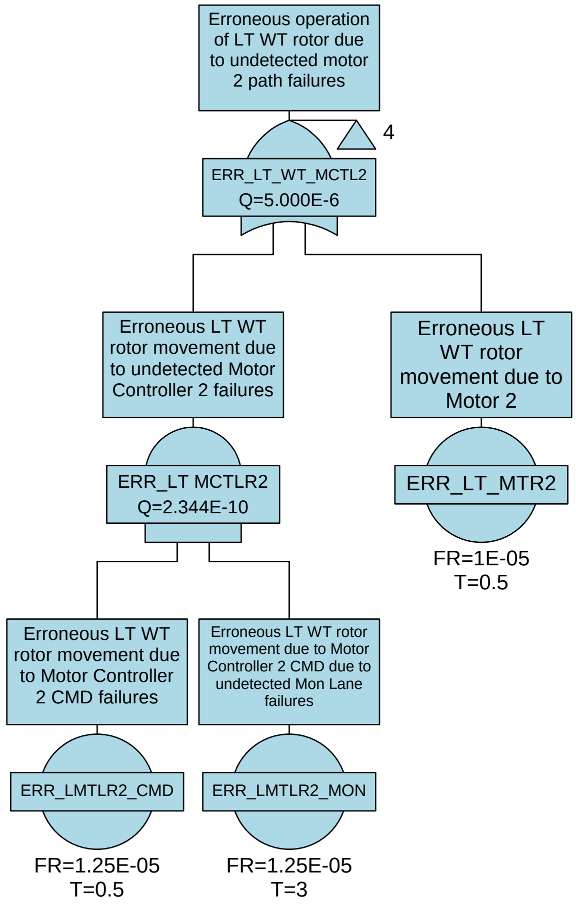
eCRM-001
FTA


FCC 1 (Blue) Erroneous or FCC 2 (Red) Erroneous but not both
2,4,5
ERR_FCC1-FCC2
Q=8.893E-11
FCC 1 or FCC 2 erroneous due to undetected failures
ERR_FCC_BOTH
Q=1.779E-10
Only 1 FCC active - Erroneous from one or the other but not both
ONE_ACTIVE
Q=0.5
FCC 1 (Blue) erroneous due to undetected failures
ERR_FCC1
Q=8.893E-11
FCC 2 (Red) erroneous due to undetected failures
ERR_FCC2
Q=8.893E-11
Erroneous FCC 1 CMD lane due to failures
ERR_FCC1_CMD
FR=7.7E-06
T=0.5
Erroneous FCC 1 CMD Lane due to undetected MON lane failures
ERR_FCC1_MON
FR=7.7E-06
T=3
Erroneous FCC 2 CMD lane due to failures
ERR_FCC2_CMD
FR=7.7E-06
T=0.5
Erroneous FCC 2 CMD Lane due to undetected MON lane failures
ERR_FCC2_MON
FR=7.7E-06
T=3

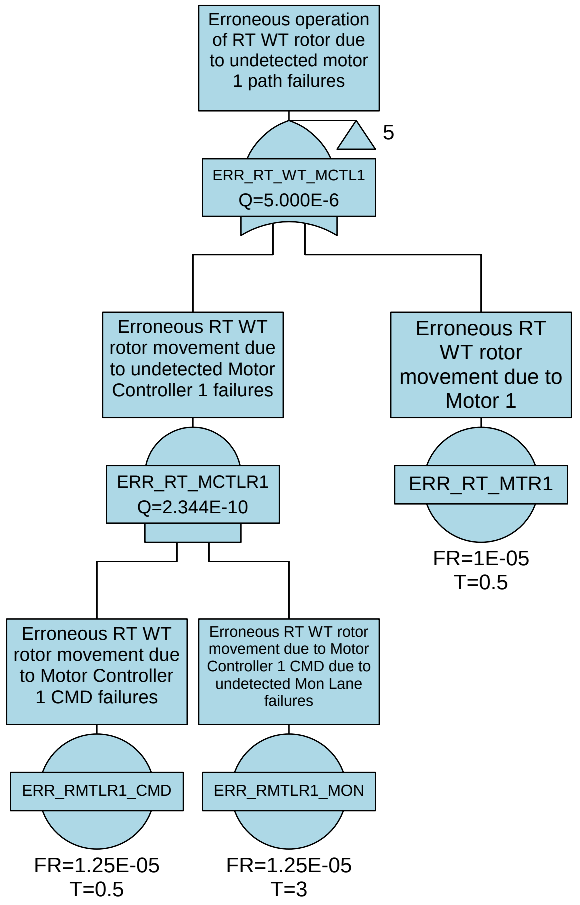
eCRM-001
FTA


Erroneous operation of LT WT rotor due to undetected motor 1 path failures
4
ERR_LT_WT_MCTL1
Q=5.000E-6
Erroneous LT WT rotor movement due to undetected Motor Controller 1 failures
ERR_LT_MCTLR1
Q=2.344E-10
Erroneous LT WT rotor movement due to Motor 1
ERR_LT_MTR1
FR=1E-05
T=0.5
Erroneous LT WT rotor movement due to Motor Controller 1 CMD failures
ERR_LMTLR1_CMD
FR=1.25E-05
T=0.5
Erroneous LT WT rotor movement due to Motor Controller 1 CMD due to undetected Mon Lane failures
ERR_LMTLR1_MON
FR=1.25E-05
T=3

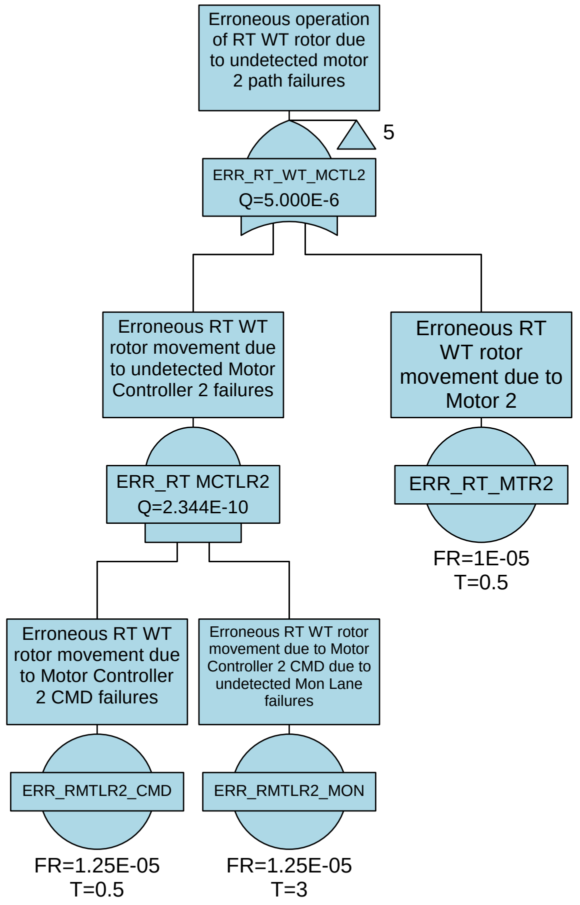
eCRM-001
FTA


Erroneous operation of LT WT rotor due to undetected motor 2 path failures
4
ERR_LT_WT_MCTL2
Q=5.000E-6
Erroneous LT WT rotor movement due to undetected Motor Controller 2 failures
Erroneous LT WT rotor movement due to Motor 2
ERR_LT MCTLR2
Q=2.344E-10
ERR_LT_MTR2
FR=1E-05
T=0.5
Erroneous LT WT rotor movement due to Motor Controller 2 CMD failures
Erroneous LT WT rotor movement due to Motor Controller 2 CMD due to undetected Mon Lane failures
ERR_LMTLR2_CMD
ERR_LMTLR2_MON
FR=1.25E-05
T=0.5
FR=1.25E-05
T=3

eCRM-001
FTA


Erroneous operation of RT WT rotor due to undetected motor 1 path failures
5
ERR_RT_WT_MCTL1
Q=5.000E-6
Erroneous RT WT rotor movement due to undetected Motor Controller 1 failures
ERR_RT_MCTLR1
Q=2.344E-10
Erroneous RT WT rotor movement due to Motor 1
ERR_RT_MTR1
FR=1E-05
T=0.5
Erroneous RT WT rotor movement due to Motor Controller 1 CMD failures
ERR_RMTLR1_CMD
FR=1.25E-05
T=0.5
Erroneous RT WT rotor movement due to Motor Controller 1 CMD due to undetected Mon Lane failures
ERR_RMTLR1_MON
FR=1.25E-05
T=3

eCRM-001
FTA


Erroneous operation of RT WT rotor due to undetected motor 2 path failures
5
ERR_RT_WT_MCTL2
Q=5.000E-6
Erroneous RT WT rotor movement due to undetected Motor Controller 2 failures
ERR_RT MCTLR2
Q=2.344E-10
Erroneous RT WT rotor movement due to Motor 2
ERR_RT_MTR2
FR=1E-05
T=0.5
Erroneous RT WT rotor movement due to Motor Controller 2 CMD failures
ERR_RMTLR2_CMD
FR=1.25E-05
T=0.5
Erroneous RT WT rotor movement due to Motor Controller 2 CMD due to undetected Mon Lane failures
ERR_RMTLR2_MON
FR=1.25E-05
T=3

A3.6.1 FPCS PSSA – RTA Concept Process

The application of the RTA concept would not alter or change the PSSA results presented in A3.6. Additional derived requirements may be captured in addition to those depicted in the example artifact depending upon the specific details of the planned implementation.

A3.7 Validate System Level Requirements and Architecture

The baseline development process will apply and capture artifacts consistent with a development plan to establish that the system and sub-system level requirements are valid, complete and correct. The development plan captures a tailored set of activities derived from ARP4754A [22] to accomplish this task.

A3.7.1 Validate RTA System Requirements

The development plan for the RTA approach would contain similar tailored activities to validate the requirements. Additionally, the RTA monitoring mechanisms must undergo a comprehensive set of activities to understand and establish that the RTA monitoring boundaries are the correct boundaries and that the definitions are complete.

Validating the RTA system requirements developed in A3.5.1 involves:

1. Validate that the simulation models and tools used in the RTA design process are of sufficient accuracy to provide reliable quantitative results. These kinds of simulations are commonly used in the design of aircraft and control systems and will typically be used extensively for the control path that is used by the human pilot. The RTA design process is constructed to ensure that these simulation capabilities are used when constructing the RTA switching boundary. Thus, the inclusion of the low-confidence control path and RTA components should not significantly complicate this validation activity.
2. Validate that all states in the identified Type I Safety Region are, in fact, safe operating points of the aircraft when the high-confidence control path is active. As previously discussed, for the eCRM-001 example, this same control path is active when the human pilot is manually flying the vehicle. Thus, this validation activity will need to be performed regardless of whether the low-confidence path was included in the design. The presence of the low-confidence path and the RTA components do not alter this activity.
3. Validate that the approximation for the switching boundary $h(s)$ that is produced using Eqs. 20 and 21 correctly classifies each vehicle state that is outside the Type III Safety Region. That is, we must ensure that, for any state s from which the low-confidence controller could drive the aircraft to a non-recoverable state, the approximation $h(s) \leq 0$. Typically, this will be validated by the use of large-scale simulation studies employing the validated simulation tools described in Step 1, above. Note that the construction of $h(s)$ will itself involve a considerable amount of simulation data and that, by construction, $h(s) \leq 0$ for all observed non-recoverable states in this dataset. This validation step will involve non-overlapping simulation data, with the selection of initial states based on competent engineering judgment, with a special focus on states that are near the approximated Type III Safety Boundary.

A3.8 Allocate System to Items

Each FCC features a high-integrity, self-monitored Digital Computation function to ensure integrity of calculation results and positive identification of channel failures. The self-monitoring is accomplished via conventional Command Lane and Monitor Lane calculation comparisons.

FCC 1 provides transmit and receive of the “Blue” communication channel data while FCC 2 provides transmit and receive of “Red” communication channel. Each FCC operates in receive only mode on the opposite FCC communication channel.

The FCCs (Blue and Red) have the internal architecture presented in Figure A-46. The FCCs have a fail-operational active/stand-by redundancy configuration. Active control is provided over each FCC’s primary system data bus. Each FCC primary system data bus is “wrap-around” monitored to ensure high-integrity control messaging to the distributed controllers. The secondary data bus into each FCC is used for management of redundant data as well as ensuring the standby FCC is ready to transition to active status at any time.

Each FCC is internally self-monitored for integrity of operation and will disengage when erroneous behaviors are identified. If either Command or Monitor Lane set of electronics fails then all messaging from that FCC terminates and the active FCC will then “pass” control to the FCC acting in backup.

Editor’s Note: This implementation strategy is consistent with many current flight control system implementations.

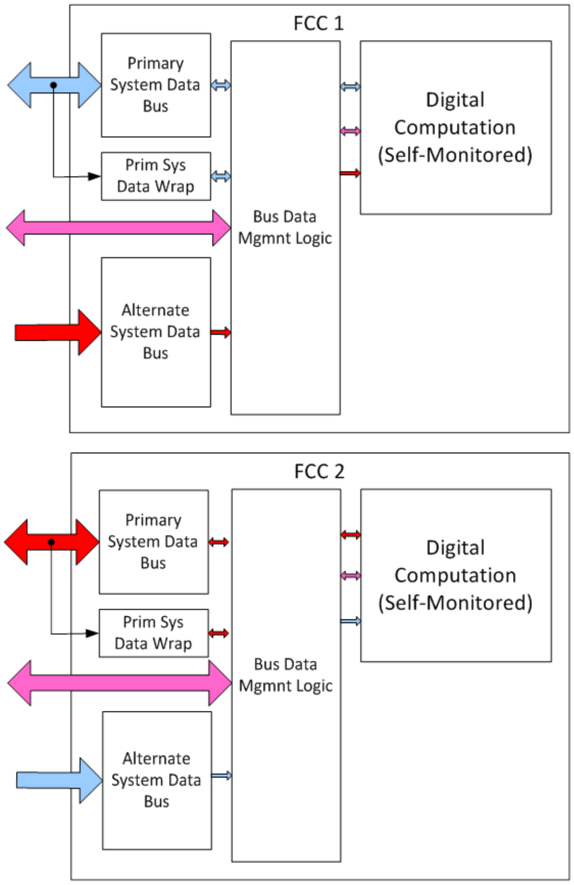


Figure A-46 eCRM-001 FCC Internal Architecture Block Diagram

A3.8.1 Baseline System Allocations

The FCC hosted airborne software will be allocated and partitioned as presented in Table A-7 Flight-Propulsion Control System FCC Software Partitioning Plan. An operating system with its associated hardware abstraction layer software will be used to host the necessary FPCS applications.

Editor's Note: The operating system must provide acceptable time partition management and memory access partition management characteristics between software developed to different levels of assurance.

Flight and propulsion control function applications (the focus of this example) are highlighted in green in Table A-7. The Input Signal Processing, Redundancy and Mode Management, Flight-Propulsion Control, and Output Processing software functions will be developed to IDAL C.

The Maintenance partition may be developed to Level D or E, depending upon safety characteristics identified in the PSSA. The Maintenance partition is enabled by the operating system and Redundancy/Mode Management functions using safety criteria.

Editor's Note: Section A2.7 highlighted that, in general, the FPCS software would all be developed to Level C, however, with appropriate interlocks within a higher level of confidence software partition, lower level of confidence software may be developed and executed.

Table A-7 Flight-Propulsion Control System FCC Software Partitioning Plan

Partition	CMD Lane Item Development Assurance Level (IDAL)	MON Lane Item Development Assurance Level (IDAL)
Boot/System Start-up-Shutdown	C	C
Operating System	C	C
Input Signal Processing	C	C
Redundancy/Mode Management	C	C
Flight-Propulsion Control	C	C
Output Processing	C	C
Maintenance	D	

The high level system software functions, depicted in Figure A-47 eCRM-001 Baseline FCC Software Function Architecture are further decomposed into the sub-functions captured in Table A-8.

Table A-8 Baseline SW Sub-Functions

Operating System • Boot, • Hardware abstraction, • Time management, • Space management	**Flight-Propulsion Control** • TRC-ACAH, • Autoland, • Mixing
Input Signal Processing • ADC 1 and ADC 2, • Pilot Command Data, • AHRS 1 and AHRS 2, • AOA 1 and AOA 2	**Redundancy/Mode Management** • Signal select and monitoring (SSM), • Normal/Degraded Mode management, • Active/Standby Mode management, • COM-MON Command Comparison, • Data bus monitoring and management
Output Processing • Command output management, • Data bus message formulation	**Maintenance** • Failure Diagnosis, • Failure Isolation

The baseline system process now “hands off” the functional and implementation requirements associated with the system and sub-system to the lower layer processes for implementation. As discussed in section A2.7, the baseline airborne electronics development will be accomplished per a tailored DO-254 [26] process and the software development will be accomplished by a tailored DO-178C/DO-331 [25][27] process.

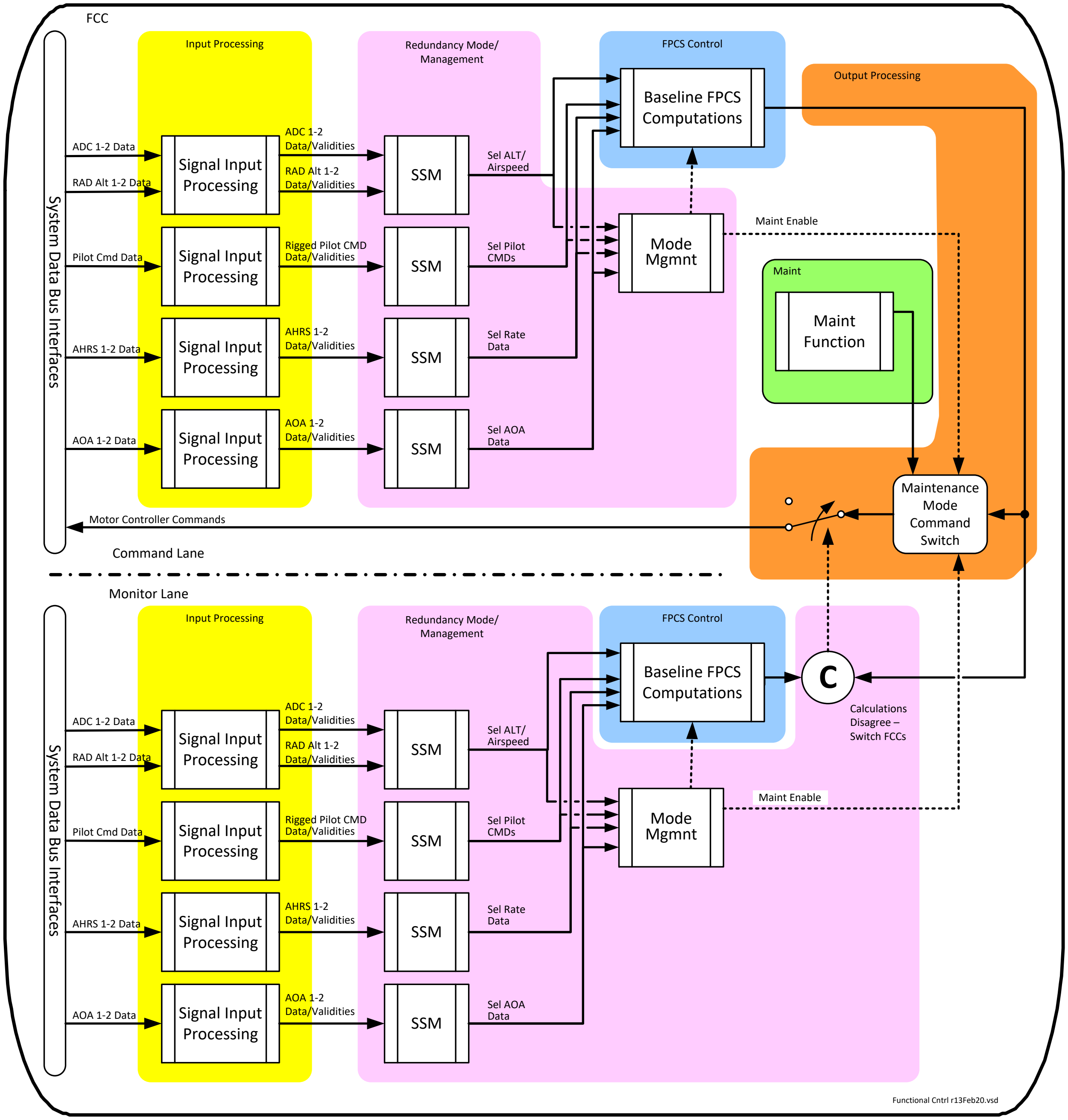


Figure A-47 eCRM-001 Baseline FCC Software Function Architecture

A3.8.2 Allocate RTA Functions to Items – Alternate Approach

The RTA self-monitoring architecture concept is presented in Figure A-48.

The RTA architecture has two different monitoring mechanisms. Similar to the baseline approach, a pair of high confidence simplified control function applications execute on two independent computer platforms and the computation results are compared. Calculations which compare indicate that no computation errors have been detected and the outputs of the FCC have high integrity.

The second monitoring mechanism involves the RTA Monitor. The RTA Monitor evaluates current aircraft response characteristics and evaluates them against predetermined safe state boundary conditions. When the current or predicted vehicle situation exceeds the predetermined safety condition, a reversion from the low-confidence executing application to a high confidence, limited control functionality capability is accomplished.

Editor's Note: The Reversionary Control capability shown in Figure A-48 is designed to provide the minimum control capability desired for the piloted vehicle. This could be rudimentary or elaborate based on vehicle manufacturer's requirements. If or when an RTA Monitor threshold is breached, mode annunciation will be provided and control commands will be faded such that vehicle responses are transient free.

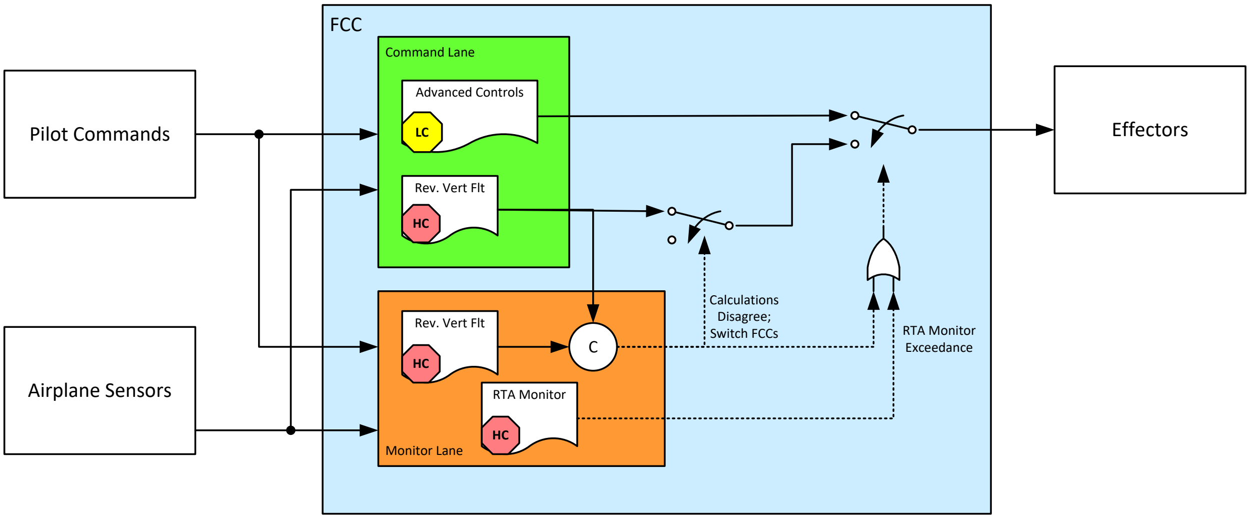


Figure A-48 RTA Control Architecture

The FCC hosted airborne RTA software will be partitioned as presented in Table A-9. An operating system with its associated hardware abstraction layer software will be used to host the necessary FPCS applications.

Editor's Note: The operating system must again provide acceptable time partition management and memory access partition management characteristics between software developed to different levels of assurance.

Flight and propulsion control function applications (the focus of this example) are highlighted in green in Table A-9 The Input Signal Processing, Redundancy and Mode Management, Flight-Propulsion Control - Reversion, and Output Processing software functions will be developed to IDAL C.

The Advanced FPC may be developed to any development assurance. In Table A-9, a tentative assignment of IDAL D has been assessed.

The Maintenance partition will be developed to Level D. The Maintenance partition is enabled by the operating system and Redundancy/Mode Management functions using safety criteria.

Table A-9 FPCS RTA FCC Software Partitioning Plan

Partition	CMD Lane Item Development Assurance Level (IDAL)	MON Lane Item Development Assurance Level (IDAL)
Boot/System Start-up-Shutdown	C	C
Operating System	C	C
Input Signal Processing	C	C
Redundancy/Mode Management	C	C
Flight-Propulsion Control – Reversion (i.e. Degraded Mode)	C	C
Flight-Propulsion Control – Advanced	D	-
RTA Monitor	-	C
Output Processing	C	C
Maintenance	D	

The FCC RTA Software Function Architecture, shown in Figure A-49, is further decomposed into the sub-functions captured in Table A-10.

Table A-10 RTA SW Sub-Functions

Operating System • Boot, • Hardware abstraction, • Time management, • Space management	**Flight-Propulsion Control** • Advanced Controls - Path Planner, - Autoland, TRC-ACAH • Reversionary Controls - TRC-ACAH, • Mixing
Input Signal Processing • ADC 1 and ADC 2, • Pilot Command Data, • AHRS 1 and AHRS 2, • AOA 1 and AOA 2	**Redundancy/Mode Management** • Signal select and monitoring (SSM), • Normal/Degraded Mode management, • Active/Standby Mode management, • COM-MON Command Comparison, • Data bus monitoring and management • RTA Monitoring
Output Processing • Command output management, • Data bus message formulation	**Maintenance** • Failure Diagnosis, • Failure Isolation

The system process now “hands off” the functional and implementation requirements associated with the system and sub-system to the lower layer processes for implementation. As discussed in section A2.7, the airborne electronics development will be accomplished per a tailored DO-254 [26] process and the software development will be accomplished by a tailored DO-178C/DO-331 [25][27] process.

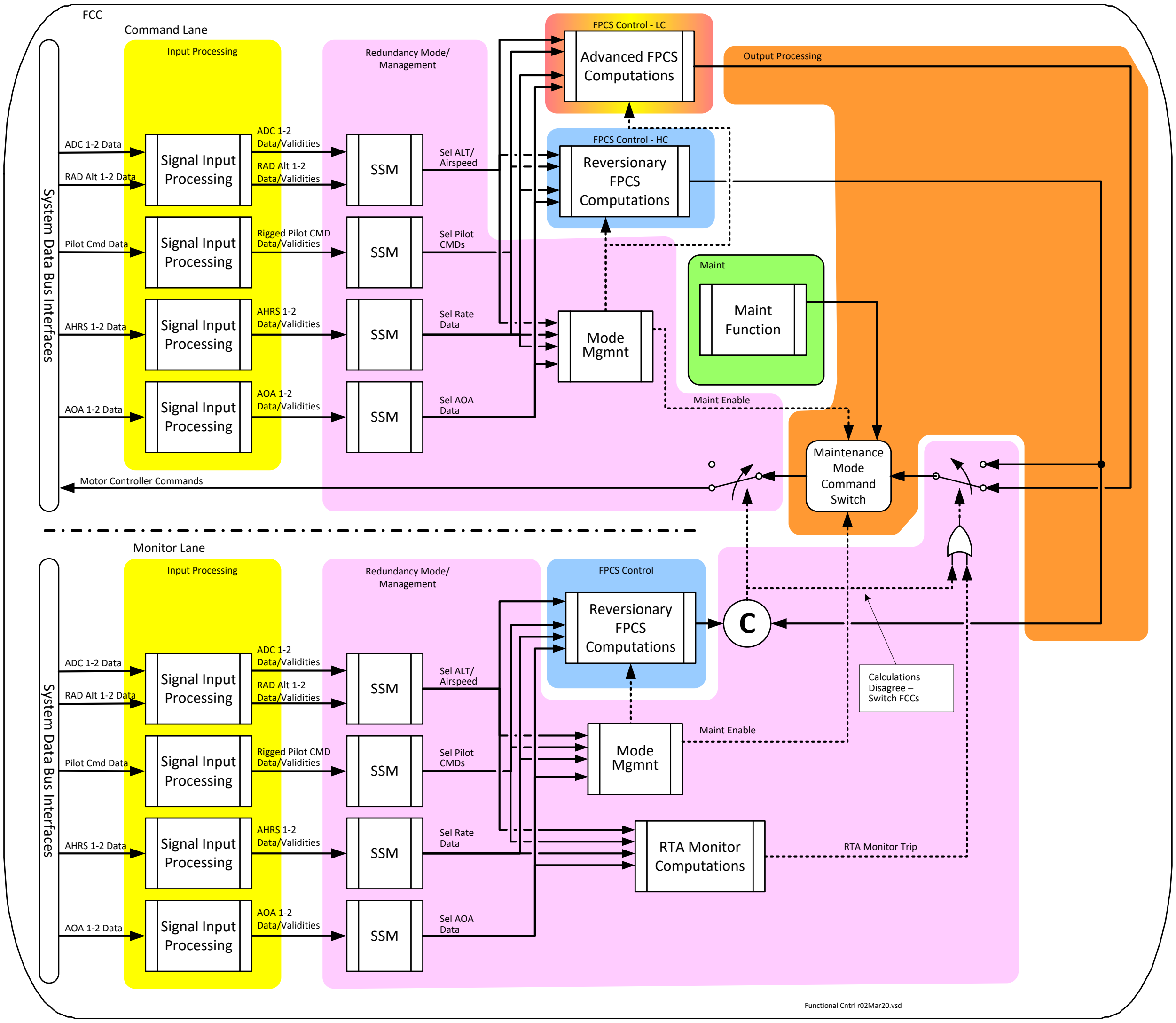


Figure A-49 eCRM-001 RTA FCC Software Function Architecture

A4.0 Airborne Electronics & Software Implementation

Editor's Note: This section describes the activities accomplished to implement the baseline and RTA approaches.

A4.1 Implement Airborne Electronics

Editor's Note: The development of the airborne electronic hardware, for either the baseline or RTA approaches is envisioned to be identical and accomplished per the development plan to IDAL C as presented in A2.7. Either development will accomplish 11 DO-254 [26] activities.

Editor's Note: The airborne electronic hardware will be sized with the computing throughput necessary to execute the software functionality allocated.

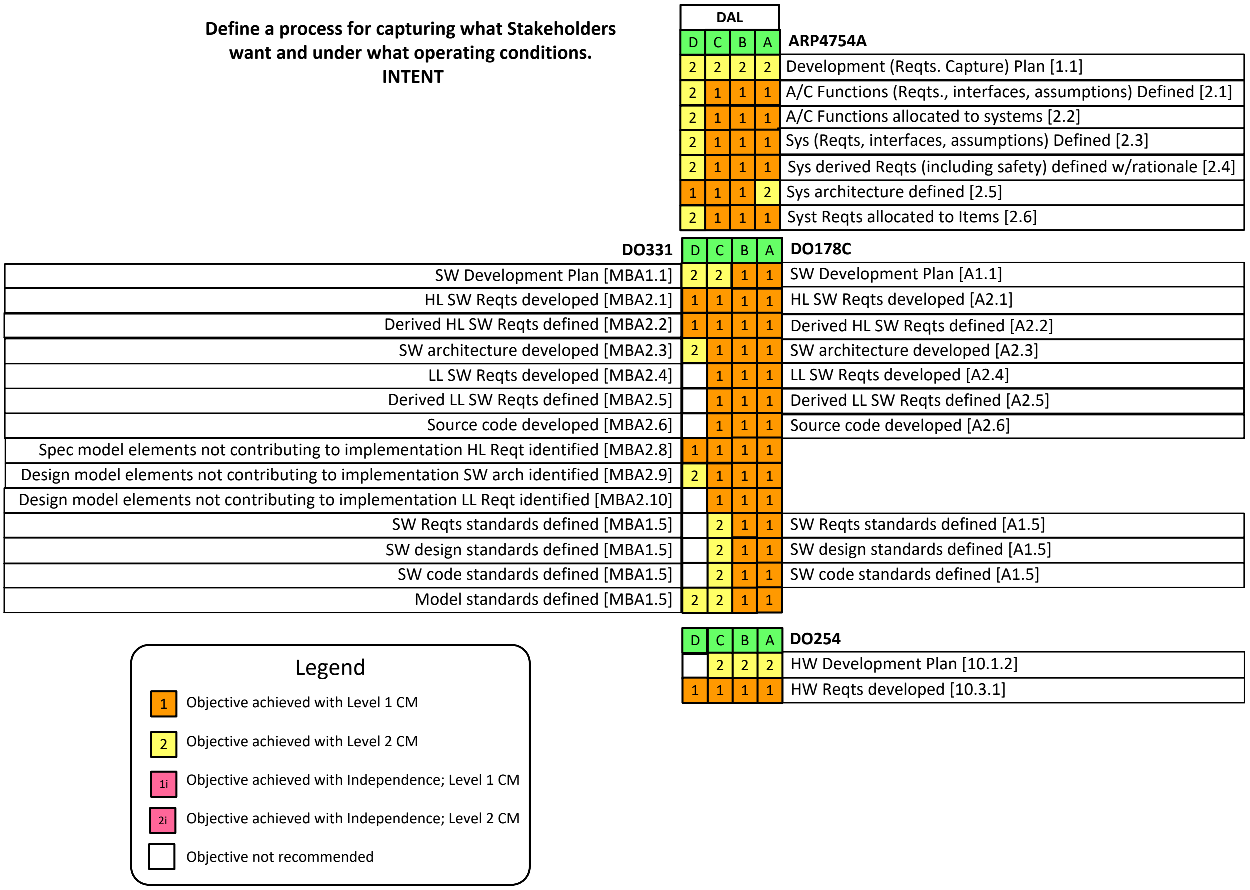
A4.2 Implement Airborne Software

Airborne software for the baseline or RTA architecture approaches will be developed using the activities identified in the defined industry process DO-178C [25] or DO-331 [27] per the levels identified in section A3.8.

A4.2.1 Implement Airborne Software – Baseline

Editor's Note: The development of the airborne software for the baseline architecture approach is accomplished per the development plan to DO-178C/DO-331 IDAL C as presented in A2.7. The activities associated with DO-178C/DO-331 [25][27] software development are generalized into four areas:

1. *Establish function behavioral intent (see* Figure A-50, *DAL column C),*
2. *Validate function behavioral intent (see* Figure A-51, *DAL column C),*
3. *Verify implementation satisfies behavioral intent (see* Figure A-52*, DAL column C),*
4. *Ensure development process correctness (see* Figure A-53*, DAL column C).*



Define a process for capturing what Stakeholders want and under what operating conditions.
INTENT

DAL				
D	C	B	A	**ARP4754A**
2	2	2	2	Development (Reqts. Capture) Plan [1.1]
2	1	1	1	A/C Functions (Reqts., interfaces, assumptions) Defined [2.1]
2	1	1	1	A/C Functions allocated to systems [2.2]
2	1	1	1	Sys (Reqts, interfaces, assumptions) Defined [2.3]
2	1	1	1	Sys derived Reqts (including safety) defined w/rationale [2.4]
1	1	1	2	Sys architecture defined [2.5]
2	1	1	1	Syst Reqts allocated to Items [2.6]

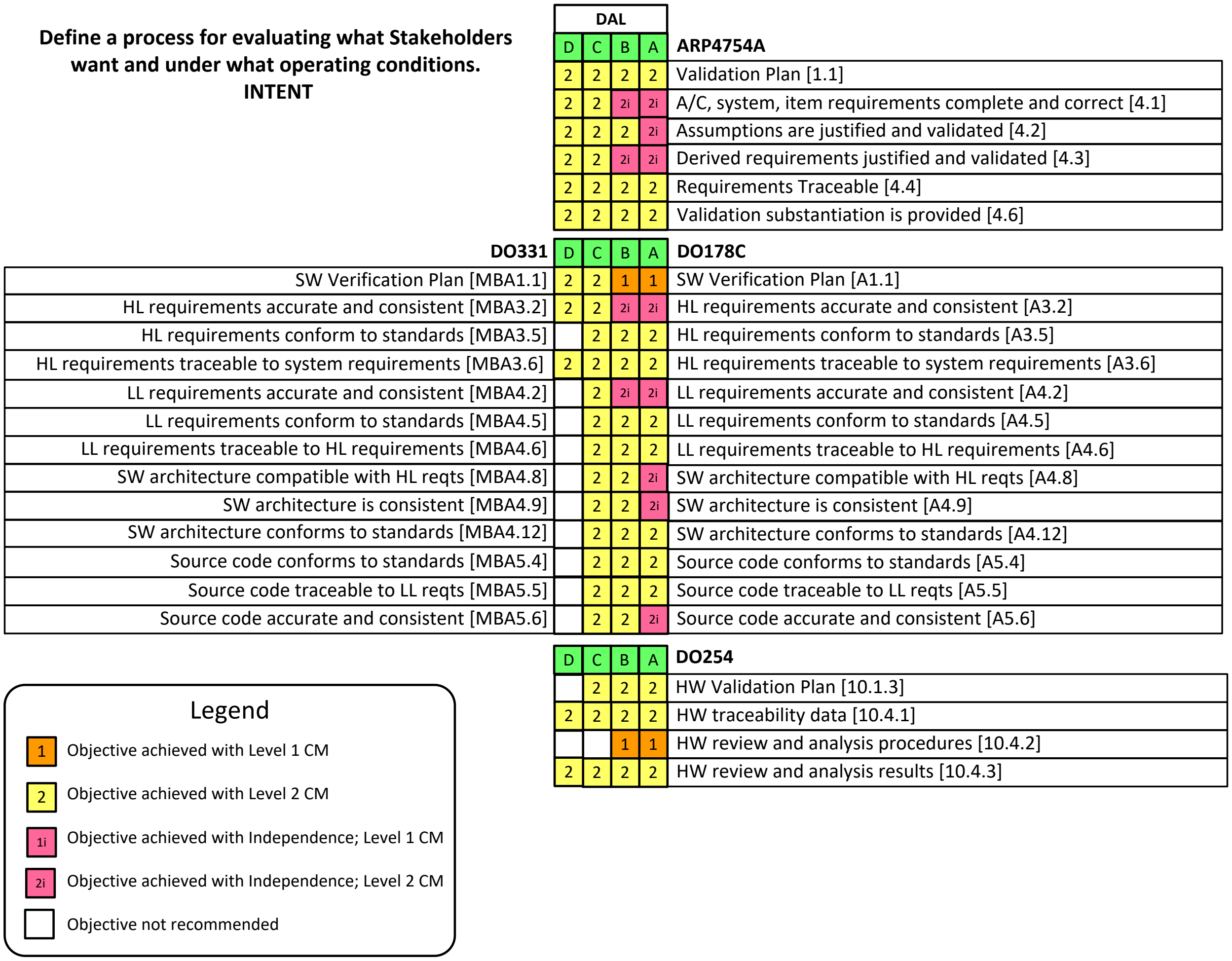
DO331	D	C	B	A	**DO178C**
SW Development Plan [MBA1.1]	2	2	1	1	SW Development Plan [A1.1]
HL SW Reqts developed [MBA2.1]	1	1	1	1	HL SW Reqts developed [A2.1]
Derived HL SW Reqts defined [MBA2.2]	1	1	1	1	Derived HL SW Reqts defined [A2.2]
SW architecture developed [MBA2.3]	2	1	1	1	SW architecture developed [A2.3]
LL SW Reqts developed [MBA2.4]		1	1	1	LL SW Reqts developed [A2.4]
Derived LL SW Reqts defined [MBA2.5]		1	1	1	Derived LL SW Reqts defined [A2.5]
Source code developed [MBA2.6]		1	1	1	Source code developed [A2.6]
Spec model elements not contributing to implementation HL Reqt identified [MBA2.8]	1	1	1	1	
Design model elements not contributing to implementation SW arch identified [MBA2.9]	2	1	1	1	
Design model elements not contributing to implementation LL Reqt identified [MBA2.10]		1	1	1	
SW Reqts standards defined [MBA1.5]		2	1	1	SW Reqts standards defined [A1.5]
SW design standards defined [MBA1.5]		2	1	1	SW design standards defined [A1.5]
SW code standards defined [MBA1.5]		2	1	1	SW code standards defined [A1.5]
Model standards defined [MBA1.5]	2	2	1	1	

D	C	B	A	**DO254**
	2	2	2	HW Development Plan [10.1.2]
1	1	1	1	HW Reqts developed [10.3.1]

Legend

1	Objective achieved with Level 1 CM
2	Objective achieved with Level 2 CM
1i	Objective achieved with Independence; Level 1 CM
2i	Objective achieved with Independence; Level 2 CM
□	Objective not recommended

Figure A-50 SW Activities for Establish Behavioral Intent



Define a process for evaluating what Stakeholders want and under what operating conditions. INTENT

DAL				
D	C	B	A	**ARP4754A**
2	2	2	2	Validation Plan [1.1]
2	2	2i	2i	A/C, system, item requirements complete and correct [4.1]
2	2	2	2i	Assumptions are justified and validated [4.2]
2	2	2i	2i	Derived requirements justified and validated [4.3]
2	2	2	2	Requirements Traceable [4.4]
2	2	2	2	Validation substantiation is provided [4.6]

DO331	D	C	B	A	**DO178C**
SW Verification Plan [MBA1.1]	2	2	1	1	SW Verification Plan [A1.1]
HL requirements accurate and consistent [MBA3.2]	2	2	2i	2i	HL requirements accurate and consistent [A3.2]
HL requirements conform to standards [MBA3.5]		2	2	2	HL requirements conform to standards [A3.5]
HL requirements traceable to system requirements [MBA3.6]	2	2	2	2	HL requirements traceable to system requirements [A3.6]
LL requirements accurate and consistent [MBA4.2]		2	2i	2i	LL requirements accurate and consistent [A4.2]
LL requirements conform to standards [MBA4.5]		2	2	2	LL requirements conform to standards [A4.5]
LL requirements traceable to HL requirements [MBA4.6]		2	2	2	LL requirements traceable to HL requirements [A4.6]
SW architecture compatible with HL reqts [MBA4.8]		2	2	2i	SW architecture compatible with HL reqts [A4.8]
SW architecture is consistent [MBA4.9]		2	2	2i	SW architecture is consistent [A4.9]
SW architecture conforms to standards [MBA4.12]		2	2	2	SW architecture conforms to standards [A4.12]
Source code conforms to standards [MBA5.4]		2	2	2	Source code conforms to standards [A5.4]
Source code traceable to LL reqts [MBA5.5]		2	2	2	Source code traceable to LL reqts [A5.5]
Source code accurate and consistent [MBA5.6]		2	2	2i	Source code accurate and consistent [A5.6]

D	C	B	A	**DO254**
	2	2	2	HW Validation Plan [10.1.3]
2	2	2	2	HW traceability data [10.4.1]
		1	1	HW review and analysis procedures [10.4.2]
2	2	2	2	HW review and analysis results [10.4.3]

Legend

1	Objective achieved with Level 1 CM
2	Objective achieved with Level 2 CM
1i	Objective achieved with Independence; Level 1 CM
2i	Objective achieved with Independence; Level 2 CM
□	Objective not recommended

Figure A-51 SW Activities to Validate Behavioral Intent

Define a process for establishing that implementation satisfies captured INTENT under operation conditions.
CORRECTNESS

DAL				
D	C	B	A	**ARP4754A**
2	2	2	2	Development Verification Plan [1.1]
2	2	1	2i	Demonstrates intended function [5.2]
2	2	1	2i	Provides confidence no unintended impacts to safety [5.2]
2	2	2	2	Implementation complies with A/C & system requirements [5.3]
2	2	2i	2i	Safety requirements are verified [5.4]
2	2	2	2	Verification substantiation is provided [5.5]
2	2	2	2	Deficiencies assessed and safety impact identified [5.6]

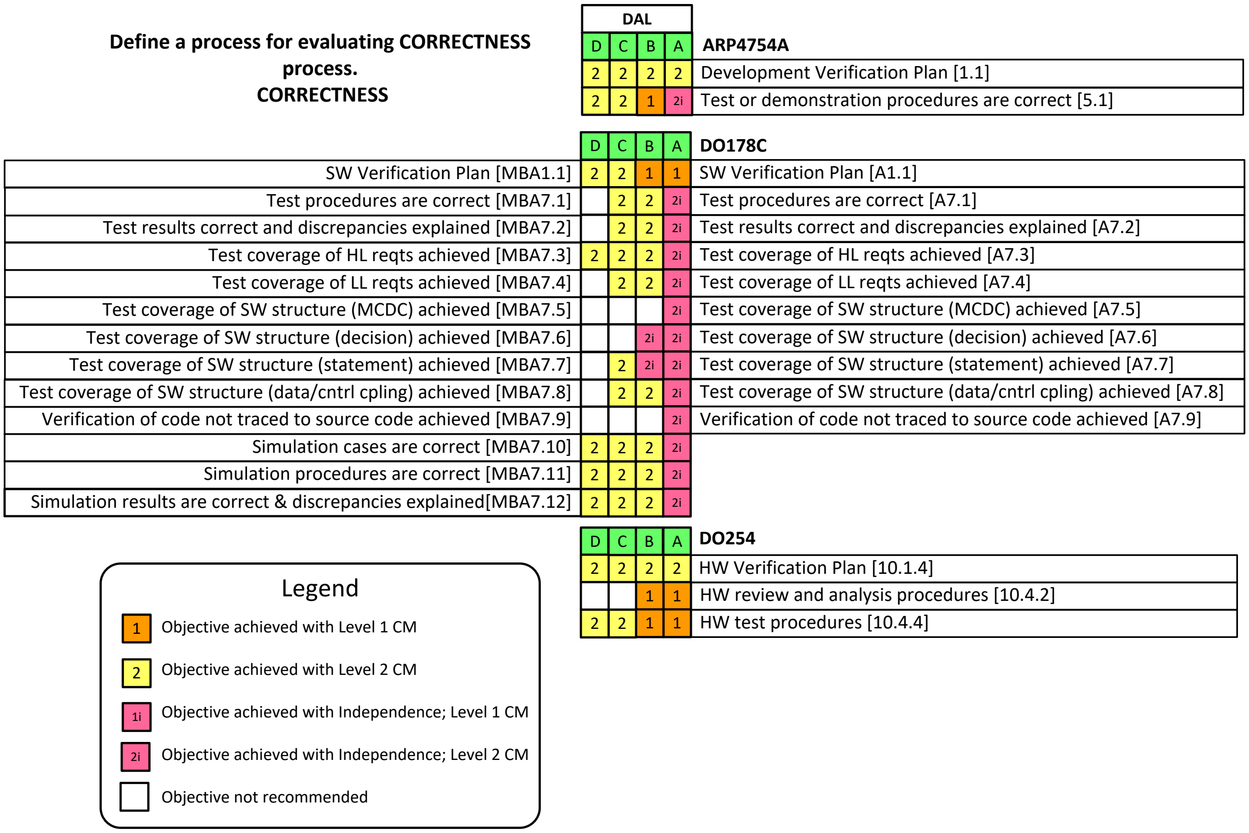
DO331	D	C	B	A	**DO178C**
SW Verification Plan [MBA1.1]	2	2	1	1	SW Verification Plan [A1.1]
Exec Obj Code & parameter data item loaded on target [MBA2.7]	1	1	1	1	Exec Obj Code & parameter data item loaded on target [A2.7]
HL reqts comply with system reqts [MBA3.1]	2	2	2i	2i	HL reqts comply with system reqts [A3.1]
HL reqts compatible with target [MBA3.3]			2	2	HL reqts compatible with target [A3.3]
HL reqts verifiable [MBA3.4]		2	2	2	HL reqts verifiable [A3.4]
Algorithms are accurate [MBA3.7]		2	2i	2i	Algorithms are accurate [A3.7]
LL reqts comply with HL reqts [MBA4.1]		2	2i	2i	LL reqts comply with HL reqts [A4.1]
LL reqts compatible with target [MBA4.3]			2	2	LL reqts compatible with target [A4.3]
LL reqts verifiable [MBA4.4]			2	2	LL reqts verifiable [A4.4]
Algorithms are accurate [MBA4.7]		2	2i	2i	Algorithms are accurate [A4.7]
SW architecture compatible with target [MBA4.10]			2	2	SW architecture compatible with target [A4.10]
SW architecture verifiable [MBA4.11]			2	2	SW architecture verifiable [A4.11]
SW partitioning integrity is confirmed [MBA4.13]	2	2	2	2	SW partitioning integrity is confirmed [A4.13]
Source code complies with LL reqts [MBA5.1]		2	2i	2i	Source code complies with LL reqts [A5.1]
Source code complies with SW architecture [MBA5.2]		2	2	2i	Source code complies with SW architecture [A5.2]
Source code is verifiable [MBA5.3]			2	2	Source code is verifiable [A5.3]
SW integration process complete & correct [MBA5.7]		2	2	2	SW integration process complete & correct [A5.7]
Parameter data item file correct & complete [MBA5.8]	2	2	2i	2i	Parameter data item file correct & complete [A5.8]
Parameter data item file verified [MBA5.9]		2	2i	2i	Parameter data item file verified [A5.9]
Exec Obj Code complies with HL reqts [MBA6.1]	2	2	2	2	Exec Obj Code complies with HL reqts [A6.1]
Exec Obj Code is robust with HL reqts [MBA6.2]	2	2	2	2	Exec Obj Code is robust with HL reqts [A6.2]
Exec Obj Code complies with LL reqts [MBA6.3]		2	2i	2i	Exec Obj Code complies with LL reqts [A6.3]
Exec Obj Code is robust with LL reqts [MBA6.4]		2	2	2i	Exec Obj Code is robust with LL reqts [A6.4]
Exec Obj Code is compatible with target [MBA6.5]	2	2	2	2	Exec Obj Code is compatible with target [A6.5]

D	C	B	A	**DO254**
2	2	2	2	HW Verification Plan [10.1.4]
		1	1	HW review and analysis procedures [10.4.2]
2	2	2	2	HW review and analysis results [10.4.3]
2	2	1	1	HW test procedures [10.4.4]
2	2	2	2	HW test results [10.4.5]

Legend

1	Objective achieved with Level 1 CM
2	Objective achieved with Level 2 CM
1i	Objective achieved with Independence; Level 1 CM
2i	Objective achieved with Independence; Level 2 CM
□	Objective not recommended

Figure A-52 SW Activities for Implementation Verification



Define a process for evaluating CORRECTNESS process.
CORRECTNESS

DAL D	DAL C	DAL B	DAL A	ARP4754A
2	2	2	2	Development Verification Plan [1.1]
2	2	1	2i	Test or demonstration procedures are correct [5.1]

	D	C	B	A	DO178C
SW Verification Plan [MBA1.1]	2	2	1	1	SW Verification Plan [A1.1]
Test procedures are correct [MBA7.1]		2	2	2i	Test procedures are correct [A7.1]
Test results correct and discrepancies explained [MBA7.2]		2	2	2i	Test results correct and discrepancies explained [A7.2]
Test coverage of HL reqts achieved [MBA7.3]	2	2	2	2i	Test coverage of HL reqts achieved [A7.3]
Test coverage of LL reqts achieved [MBA7.4]		2	2	2i	Test coverage of LL reqts achieved [A7.4]
Test coverage of SW structure (MCDC) achieved [MBA7.5]				2i	Test coverage of SW structure (MCDC) achieved [A7.5]
Test coverage of SW structure (decision) achieved [MBA7.6]			2i	2i	Test coverage of SW structure (decision) achieved [A7.6]
Test coverage of SW structure (statement) achieved [MBA7.7]		2	2i	2i	Test coverage of SW structure (statement) achieved [A7.7]
Test coverage of SW structure (data/cntrl cpling) achieved [MBA7.8]		2	2	2i	Test coverage of SW structure (data/cntrl cpling) achieved [A7.8]
Verification of code not traced to source code achieved [MBA7.9]				2i	Verification of code not traced to source code achieved [A7.9]
Simulation cases are correct [MBA7.10]	2	2	2	2i	
Simulation procedures are correct [MBA7.11]	2	2	2	2i	
Simulation results are correct & discrepancies explained[MBA7.12]	2	2	2	2i	

D	C	B	A	DO254
2	2	2	2	HW Verification Plan [10.1.4]
		1	1	HW review and analysis procedures [10.4.2]
2	2	1	1	HW test procedures [10.4.4]

Legend

1	Objective achieved with Level 1 CM
2	Objective achieved with Level 2 CM
1i	Objective achieved with Independence; Level 1 CM
2i	Objective achieved with Independence; Level 2 CM
□	Objective not recommended

Figure A-53 SW Activities for SW Process Evaluation

The count of software development assurance activities, for both FCC Lanes in the baseline approach, is summarized in Table A-11.

Table A-11 Baseline Assurance Activity Summary

		Intent Activities	Intent Validation Activities	Correctness Activities (Verification)	Process Correctness Activities	Level C ARP Proc Activities	Level C DO-178C/ DO331 Activities
Baseline						110	880
FCC CMD Lane	**Flight-Propulson Control**	7	6	7	2	22	
	TRC-ACAH	14	13	18	10		55
	Autoland	14	13	18	10		55
	Mixing	14	13	18	10		55
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	18	10		55
	Normal/Degraded	14	13	18	10		55
	Active/Standby	14	13	18	10		55
	Data bus Mon/Mgmnt	14	13	18	10		55
	Output Processing	7	6	7	2	22	
	Cmd Output Mgmnt	14	13	18	10		55
	Data Bus Msg Formulation	14	13	18	10		55
FCC MON Lane	**Flight-Propulson Control**	7	6	7	2	22	
	TRC-ACAH	14	13	18	10		55
	Autoland	14	13	18	10		55
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	18	10		55
	Normal/Degraded	14	13	18	10		55
	Active/Standby	14	13	18	10		55
	Data bus Mon/Mgmnt	14	13	18	10		55
	COM-MON Comparision	14	13	18	10		55

A4.2.2 Implement Airborne Software – RTA

Editor's Note: The development of the airborne software for the RTA approach is accomplished per the development plan to the DO-178C/DO-331 IDAL C and D as presented in Table A-10. *The activities associated with DO-178C/DO-331 [25][27] software development are generalized into four areas:*

1. *Establish function behavioral intent (see* Figure A-50*, DAL column C & D),*
2. *Validate function behavioral intent (see* Figure A-51*, DAL column C & D),*
3. *Verify implementation satisfies behavioral intent (see* Figure A-52*, DAL column C & D),*
4. *Ensure development process correctness (see* Figure A-53*, DAL column C & D).*

The count of software development assurance activities, for both FCC Lanes in the RTA approach, is summarized in Table A-12.

Table A-12 RTA Assurance Activity Summary

		Intent Activities	Intent Validation Activities	Correctness Activities (Verification)	Process Correctness Activities	Level C ARP Proc Activities	Level C DO-178C/ DO331 Activities
RTA						110	848
FCC CMD Lane	**Flight-Propulson Control**	7	6	7	2	22	
	Autoland-TRC-ACAH	7	3	8	5		23
	Reversion (TRC-ACAH)	14	13	18	10		55
	Mixing	14	13	18	10		55
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	18	10		55
	Normal/Degraded	14	13	18	10		55
	Active/Standby	14	13	18	10		55
	Data bus Mon/Mgmnt	14	13	18	10		55
	Output Processing	7	6	7	2	22	
	Cmd Output Mgmnt	14	13	18	10		55
	Data Bus Msg Formulation	14	13	18	10		55
FCC MON Lane	**Flight-Propulson Control**	7	6	7	2	22	
	Reversion (TRC-ACAH)	14	13	18	10		55
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	18	10		55
	Normal/Degraded	14	13	18	10		55
	Active/Standby	14	13	18	10		55
	Data bus Mon/Mgmnt	14	13	18	10		55
	COM-MON Comparision	14	13	18	10		55
	RTA Monitoring	14	13	18	10		55

A5.0 System Integration – Baseline and RTA

The system integration activities associated with either assurance approach would be essentially identical. Required system functionality would be established as being implemented correctly by incrementally adding system and software functionality to the integrated system elements.

A5.1 SSA – Baseline or RTA

The FPCS final implementation process accomplishes a system safety assessment to evaluate the existing system platform and verify using multiple different tools (as necessary) to show that the final implementation achieves the safety objectives captured in requirements, SFHA and AFHA analyses.

A6.0 Development for Higher Confidence Solutions

The case study focused on the development of a function at assurance level commensurate with 14CFR Part 23 [3] regulatory requirements and guidance material. It has been noted however, that alternate regulation sets and/or special conditions may establish the need for higher levels of confidence in the resulting function solution. This section discusses some of the assurance impacts associated with the goal of achieving a higher level of confidence.

As noted in Appendix B2.2 and B2.3, certification approaches to 14CFR27 [4] or EASA VTOL-SC-01 [11] would result in changes to not only the development process assurance strategy but also to the physical implementations. Table B-5 summarizes the development assurance assignment increase to Level A, the quantitative assurance objective increase to 10^{-9} and the need to satisfy “Very High Level of Confidence” (see Table B-6) characteristics.

A6.1 Process Assurance Deltas for Higher Confidence from Case Study

The development process changes from the baseline of Level C to Level A are relatively straight forward. With the increased level of rigor associated with the change in assurance levels comes the increase in activities associated with accomplishing the system, electronic hardware and software developments. Many of the same Level C tasks are accomplished but must now be done ensuring process independence and a high level of data configuration control. Figure A-50 thru Figure A-53 Column A activities would now be accomplished and generate the associated process evidence.

The revised count of development assurance activities, for both FCC Lanes in the high confidence scenario for both baseline and RTA approaches, is summarized in Table A-13.

Table A-13 High Confidence Assurance Activity Summary

		Intent Activities	Intent Validation Activities	Correctness Activities (Verification)	Process Correctness Activities	Level A ARP Proc Activities	Level A DO-178C/ DO331 Activities
Baseline						110	1024
CMD	**Flight-Propulson Control**	7	6	7	2	22	
	TRC-ACAH	14	13	24	13		64
	Autoland	14	13	24	13		64
	Mixing	14	13	24	13		64
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	24	13		64
	Normal/Degraded	14	13	24	13		64
	Active/Standby	14	13	24	13		64
	Data bus Mon/Mgmnt	14	13	24	13		64
	Output Processing	7	6	7	2	22	
	Cmd Output Mgmnt	14	13	24	13		64
	Data Bus Msg Form	14	13	24	13		64
MON	**Flight-Propulson Control**	7	6	7	2	22	
	TRC-ACAH	14	13	24	13		64
	Autoland	14	13	24	13		64
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	24	13		64
	Normal/Degraded	14	13	24	13		64
	Active/Standby	14	13	24	13		64
	Data bus Mon/Mgmnt	14	13	24	13		64
	COM-MON Comparision	14	13	24	13		64
RTA						110	983
CMD	**Flight-Propulson Control**	7	6	7	2	22	
	Autoland-TRC-ACAH	7	3	8	5		23
	Reversion (TRC-ACAH)	14	13	24	13		64
	Mixing	14	13	24	13		64
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	24	13		64
	Normal/Degraded	14	13	24	13		64
	Active/Standby	14	13	24	13		64
	Data bus Mon/Mgmnt	14	13	24	13		64
	Output Processing	7	6	7	2	22	
	Cmd Output Mgmnt	14	13	24	13		64
	Data Bus Msg Form	14	13	24	13		64
MON	**Flight-Propulson Control**	7	6	7	2	22	
	Reversion (TRC-ACAH)	14	13	24	13		64
	Redundancy/Mode Mgmnt	7	6	7	2	22	
	SSM	14	13	24	13		64
	Normal/Degraded	14	13	24	13		64
	Active/Standby	14	13	24	13		64
	Data bus Mon/Mgmnt	14	13	24	13		64
	COM-MON Comparision	14	13	24	13		64
	RTA Monitoring	14	13	24	13		64

A6.1.1 Very HC Software and Hardware Deltas

Current 14CFR Part 25 flight control system solutions provide a starting point for the development and implementation diversity which may be required for the integrated flight-propulsion control system solution for the 14CFR Part 27 or EASA SC-01 Level A requirements.

The Command and Monitor Lane high confidence software solutions, shown now in Table A-14 for the Baseline and Table A-15 for RTA have been escalated to Level A. Note that the RTA Advance Control may still be accomplished to a low LOC since it is being protected by the VHC RTA Monitor Lane capabilities.

Table A-14 Baseline FPCS FCC HC SW Development

Partition	CMD Lane Item Development Assurance Level (IDAL)	MON Lane Item Development Assurance Level (IDAL)
Boot/System Start-up-Shutdown	A	A
Operating System	A	A
Input Signal Processing	A	A
Redundancy/Mode Management	A	A
Flight-Propulsion Control	A	A
Output Processing	A	A
Maintenance	D	

Table A-15 RTA FPCS FCC HC Software Partitioning Plan

Partition	CMD Lane IDAL	MON Lane IDAL
Boot/System Start-up-Shutdown	A	A
Operating System	A	A
Input Signal Processing	A	A
Redundancy/Mode Management	A	A
Flight-Propulsion Control – Reversion (i.e. Degraded Mode)	A	A
Flight-Propulsion Control – Advanced	D	-
RTA Monitor	-	A
Output Processing	A	A
Maintenance	D	

Additionally the certification authorities may escalate the solution to represent extremely high confidence criteria. For EHC, diversity in the software tooling used to develop the software and or different software languages, different compilers, etc. between the Command and Monitor Lane implementations may be required.

The escalation in assurance level to VHC or EHC and quantitative safety criteria also means revisions to the electronics solutions. Diversity between the Command and Monitor Lanes from a microprocessor type may also be required.

From a system architecture perspective, the development escalation may also require the addition of diverse reversionary control capability and/or the adding redundant computation pathways. Figure A-54 presents a potential RTA architecture very-high confidence approach. The RTA Monitoring mechanism would be duplicated in each of two diversely implemented (both hardware and software) computer lanes. Reversionary control is provided in each FCC and is activated if the RTA monitor exceeds a safety boundary and the Monitor Lane reversionary control results compare to those of the Command Lane. If the Monitor Lane identifies a miscompare between COM and MON reversionary control results or in the RTA Monitor results, then the FCC is failed and control is transferred to the alternate FCC.

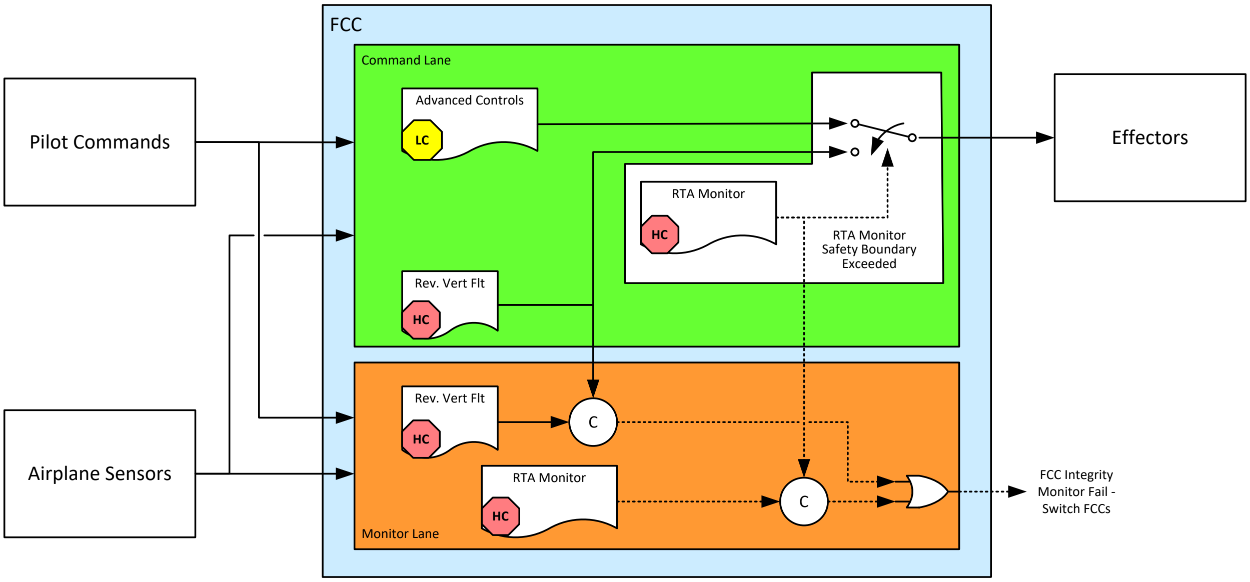


Figure A-54 FPCS FCC HC Architecture

A6.1.2 Very HC Software Reuse Scenario

When a reuse or change in Advance Control functionality is considered, the advantages of the RTA assurance concept are even more pronounced.

An update to the Advanced TRC-ACAH and Autoland functionality is contemplated. System requirements and algorithms are to be enhanced and implemented within the Baseline VHC or RTA architecture approaches. What level of process activities would be necessary to make the revised functionality?

Table A-16 captures a potential definition for the number of activities which would need to be accomplished for the two development approaches. Some high level assumptions are made in order to quantify the activities: 1) Changed functionality is defined and captured in system level requirements which flow to implementation level through requirement decomposition; 2) One “unit” of software results from the system level definitions which is then implemented, validated and verified; 3)

For the Baseline development approach, Table A-16 highlights that most of the development activities are encompassed in the Command Lane however, since the Monitor Lane is executing the same algorithms for comparison similar change activities are required. A single “improvement” results in software development activities in both Lanes. In the postulated scenario, the changed functionality may invoke the need to repeat up to 464 software development activities.

In the RTA approach, only the system activities and low confidence software development activities would be achieved. The RTA Monitoring mechanism, defined during initial development, would not need to be revised since the characteristics are based on aircraft parametrics. In the postulated scenario, the change in functionality may invoke only 60 software development activities, a reduction of up to seven (7) over the Baseline development approach.

Table A-16 VHC Reuse Activity Count

			Intent Activities	Intent Validation Activities	Correctness Activities (Verification)	Process Correctness Activities		Level A ARP Proc Activities	Level A DO-178C/ DO331 Activities
Baseline								44	464
CMD	**Flight-Propulson Control**		7	6	7	2		22	
	TRC-ACAH		14	13	24	13			64
	Autoland		14	13	24	13			64
	Mixing		14	13	24	13			64
	Redundancy/Mode Mgmnt							0	
	SSM								0
	Normal/Degraded				24				24
	Active/Standby				24				24
	Data bus Mon/Mgmnt								0
	Output Processing							0	
	Cmd Output Mgmnt				24				24
	Data Bus Msg Form								0
MON	**Flight-Propulson Control**		7	6	7	2		22	
	TRC-ACAH		14	13	24	13			64
	Autoland		14	13	24	13			64
	Redundancy/Mode Mgmnt							0	
	SSM								0
	Normal/Degraded				24				24
	Active/Standby				24				24
	Data bus Mon/Mgmnt								0
	COM-MON Comparision				24				24
RTA								22	60
CMD	**Flight-Propulson Control**		7	6	7	2		22	
	Autoland-TRC-ACAH		7	3	8	5			23
	Reversion (TRC-ACAH)								0
	Mixing				24	13			37
	Redundancy/Mode Mgmnt							0	
	SSM								0
	Normal/Degraded								0
	Active/Standby								0
	Data bus Mon/Mgmnt								0
	Output Processing							0	
	Cmd Output Mgmnt								0
	Data Bus Msg Form								0
MON	**Flight-Propulson Control**							0	
	Reversion (TRC-ACAH)								0
	Redundancy/Mode Mgmnt							0	
	SSM								0
	Normal/Degraded								0
	Active/Standby								0
	Data bus Mon/Mgmnt								0
	COM-MON Comparision								0
	RTA Monitoring								0

Appendix B: Current Industry Practice Baseline Objective Derivation

B1.0 Current Assurance Practices Baseline

The certification goal of assurance is to efficiently provide safety aspect coverage of systems and equipment providing complex and interrelated functions through:

- Development assurance using a combination of process assurance and verification coverage criteria,
- Structured analysis or assessment techniques applied at the aircraft level to integrated and interacting systems [122].

The combination of these two aspects provides increased confidence in identification and correction of errors/mistakes in requirements, design, integration or interaction effects and that the implementation satisfies both the qualitative and quantitative certification criteria. To accomplish these certification goals, each aircraft standard category contains a “safety rule” and provides guidance material for safety and assurance objectives which are accomplished to show compliance to the “safety rule”. Historically, this was the “xx-1309 regulation in each regulatory part and this regulations advisory material provided the safety and assurance criteria for that vehicle systems and equipment. The associated advisory material identified the acceptable means of compliance.

From the perspective of evaluating alternate assurance concepts, it is important to establish the appropriate rule-set in order to understand the safety objectives associated with the final implementation. Each certification regulatory set contains safety and assurance objectives for implementations based on regulatory advisory material applicable to that part. How the Applicant is to address an evaluation of the final implementation as well as the development process tailored to address the severity of error or failure consequences have been described for Applicant response.

As part of this study, we have extracted these certification criteria in order to capture a baseline set of objectives for the current practices. The current practice baseline criteria will then be used comparatively to the alternate assurance concepts studied.

B1.1 Assurance Definition

A consistent idea of assurance must first be established in order to understand what constitutes “current industry assurance practice” and enable the contrast to an alternate assurance concept. The following two industry definitions define assurance from two abstraction levels.

- Assurance – the planned and systematic actions necessary to provide adequate confidence and evidence that a product or process satisfies given requirements. (DO-178B)
- Development Assurance – All those planned and systematic actions used to substantiate, to an adequate level of confidence, that errors in requirements, design and implementation have been identified and corrected such that the system satisfies the applicable certification basis (CS-25).

The first definition for assurance is broader and encompasses any set of actions which may be applied to evaluate/establish that a product has an adequate level of confidence and evidence for satisfying the requirements. Development assurance provides only one portion of the “assurance planned and systematic actions” by defining actions associated with a development process used to substantiate and provide a level of confidence.

Any alternate assurance concept must then provide both a development assurance element as well as a technical assurance evaluation component to create adequate confidence in the final implementation satisfying it’s given requirements.

B2.0 Baseline Practice Objective Identification

Identifying the baseline practice objectives for each aircraft category requires deciphering the certification process to select an applicable “safety regulation” and certification compliance criteria.

The certification process is based on a selection of regulations applicable to the new airplane being certificated. As described in 14 CFR 21.17 [2] and presented graphically in Figure B-1, the Applicant selects applicable regulations by aircraft type (or if certifying an engine – 14 CFR Part 33 or propeller – 14 CFR Part 35). The applicable regulation set for general aviation airplanes is 14CFR Part 23 – Airworthiness Standards – Normal Category Airplane [3]. Rotorcraft regulations are defined under 14CFR Part 27 – Airworthiness Standards – Normal Category Rotorcraft [4] and 14CFR Part 29 Airworthiness Standards – Transport Category Rotorcraft [10].

The development and certification of general aviation airplane and rotorcraft systems and equipment is straight forward. The General Aviation process and objective identification is presented in section B2.1 with the Rotorcraft process and objective identification identified in section B2.2.

Vehicles which combine the features of both airplane and rotorcraft categories present a challenge to civil aviation industry, however. These vehicles blend the certification concepts using technology to merge once independent airplane and rotorcraft functions into a new integrated vehicle format.

These new vehicles, which do not correlate to one of the defined airworthiness frameworks, must create a certification project which considers the full suite of published regulations and select an applicable set to propose in the specific aircraft certification plan. The final applied regulatory set for a given vehicle will be the optimum “mix and match” of regulations from multiple parts negotiated between the Applicant and the certification authority.

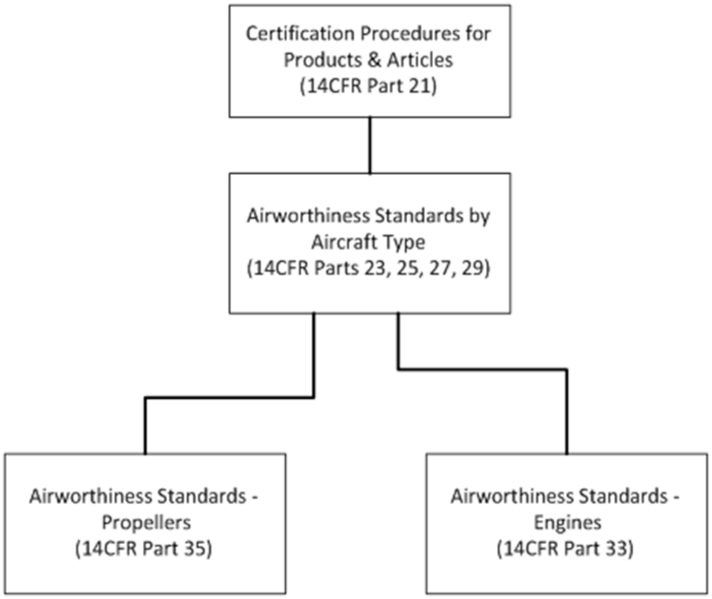

Certification Procedures for Products & Articles (14CFR Part 21)
Airworthiness Standards by Aircraft Type (14CFR Parts 23, 25, 27, 29)
Airworthiness Standards - Propellers (14CFR Part 35)
Airworthiness Standards - Engines (14CFR Part 33)

Figure B-1 Aircraft Certification Hierarchy for GA & Rotorcraft

One version of a new vehicle regulatory framework has been established by the European Aviation Safety Agency (EASA) in EASA Vertical Takeoff and Landing (VTOL) Special Condition [11]. This special condition identifies and provides some guidance on assurance objectives for a new vehicle type. The baseline objectives for VTOL vehicles are discussed in section B2.3.

B2.1 Normal Category Airplane

The FAA Normal Category Airplane regulations (14CFR Part 23 [2]) were revised in 2017 to create performance and objective based regulations. Advisory Circular (i.e. ACs) compliance criteria have not been issued to date, however acceptable means of compliance (AMOC) to the new regulations has been published in the Federal Register (Federal Register Volume 83, No 92 Notice 23-81-NOA [13] and in PS-AIR-23-09 System Level Verification of Electronic Equipment (Software and Airborne Electronic Hardware) [14]. New AMOC criteria (ASTM consensus standards) were defined in these published references.

The following ASTM standards were identified for showing 14CFR Part 23 compliance:

- F3264-18 Standard Specification for Normal Category Aeroplanes Certification [15],
- F3061-17 Standard Specification for Systems and Equipment in Small Aircraft [16],
- F3230-17 Standard Practice for Safety Assessment of Systems and Equipment in Small Aircraft [17],
- F3153-15 Standard Specification for Verification of Avionics Systems [18],
- F3309-18, Standard Practice for Simplified Safety Assessment of Systems and Equipment in Small Aircraft [20].

In the revised 14CFR Part 23 regulations, safety assurance criteria for systems and equipment has been captured in §23.2510. There are two potential assurance process approaches which may be considered:

1. Apply the Standard Specification for Systems and Equipment in Small Aircraft (F3061-17 [16]) defined process or,
2. Apply the Standard Specification for Verification of Avionics Systems (F3153-15 [18]) defined process.

B2.1.1 Approach 1- Using F3061-17 Standard for Objective Identification

Process approach 1 is summarized in Figure B-2. The 14CFR Part 23.2510 safety objective decoding process is described in the referenced F3061-17 ASTM standard. This process identifies both the quantitative and qualitative safety assurance objectives which should be satisfied for certification compliance.

The ASTM F3061-17 framework uses an Aircraft Type Code (ATC), aircraft stall speed and planned aircraft operating meteorological conditions in order to select the appropriate safety assurance activities and establish that software and hardware development must be accomplished to an appropriate development assurance level (I.E. Is ASTM section 4.2.3 applicable?).

ASTM F3061-17 Table 1 Airworthiness Level of the ATC is based on the number of passengers and crew the planned vehicle will transport (“1” -1 pax; “2” - 2-6 pax; “3” - 7-9 pax; “4” – more than 10 pax).

ASTM F3061-17 Section 4.2.3 will not be applicable to the aircraft development if the vehicle is Airworthiness Level 1, has a stall speed less than or equal to 45 knots (83km/h) and is for day VFR operating conditions.

Typical GA airworthiness level would be “2” (2-6 pax) so airworthiness level portion of the ATC would be AW II. This establishes ASTM F3016-17 Section 4.2.3 as applicable. A safety *Assessment Level* for the aircraft must now be determined per ASTM F3230-17 in order to identify the development assurance levels for software (SW) and airborne electronic hardware (AEH).

The *Assessment Level* in F3230-17 Table 3 is based on the quantity and type of combustion (reciprocating or turbine) engines planned for use. If we ignore the engine type, while focusing on the engine quantity, the assignment of *Assessment Level II* may be identified*.*

Using the *Assessment Level II* assignment in F3061-17 Table 2, results in airborne software (SW) and airborne electronic hardware (AEH) development assurance objective assignments of level “C”. Assignment of the IDAL for SW and AEH HW may also consider development architecture per ARP4754A [22] FDAL/IDAL assignment per Table 3 or for simplicity use ARP4754A Table 2 (i.e. catastrophic = A, Hazardous = B, Major = C, etc.).

Acceptable means of achieving the development objectives are identified in F3061-17 section 4.2.3.1 as DO-178C [24] for airborne software and DO-254 [26] for airborne electronic hardware.

No development assurance objective level is identified for aircraft or system level functions. However, F3061-17 implies a need to capture intended function so that verification can be accomplished to establish that final implementation provides that function when installed and operated in environmental conditions.

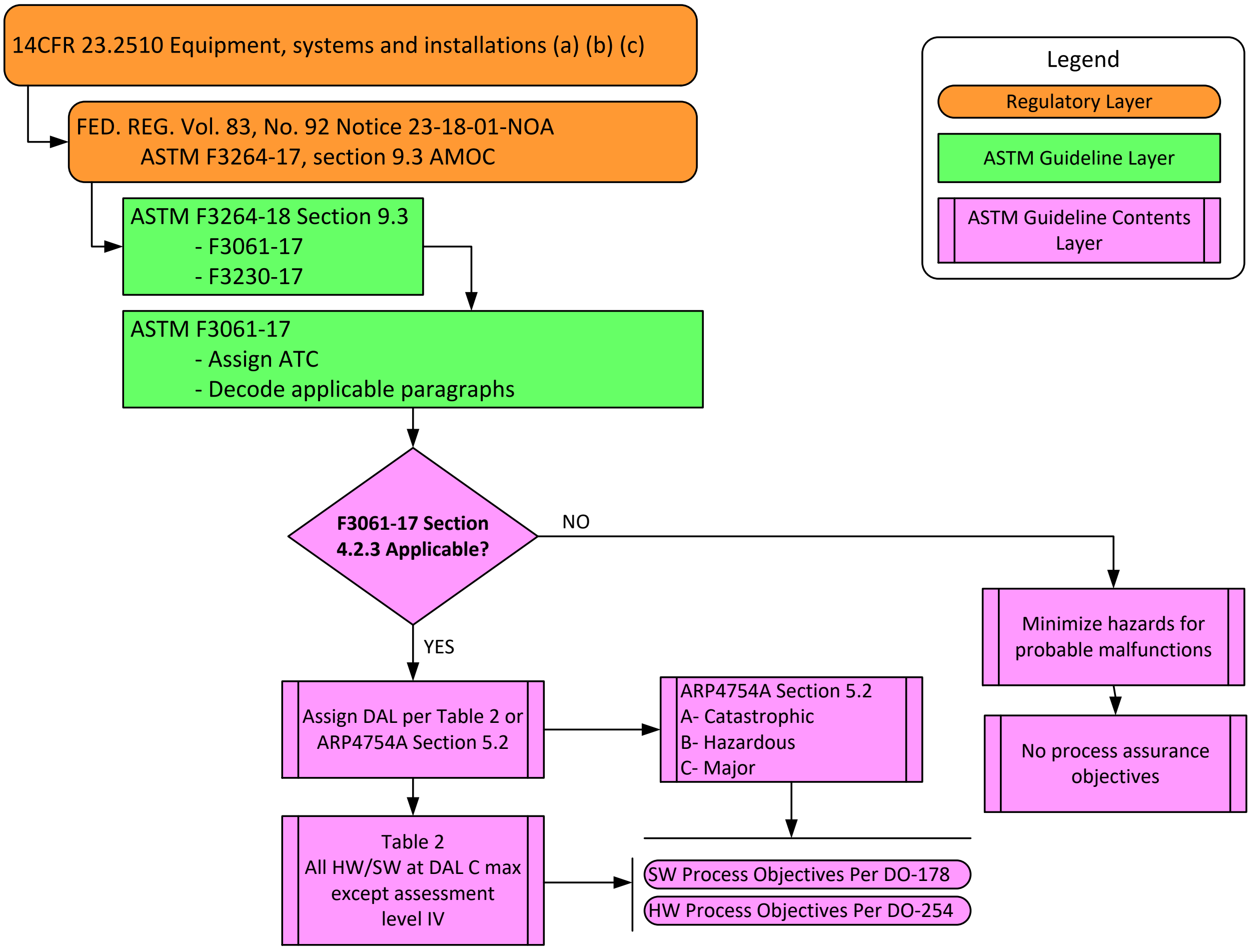

14CFR 23.2510 Equipment, systems and installations (a) (b) (c)
FED. REG. Vol. 83, No. 92 Notice 23-18-01-NOA
ASTM F3264-17, section 9.3 AMOC
ASTM F3264-18 Section 9.3
- F3061-17
- F3230-17
ASTM F3061-17
- Assign ATC
- Decode applicable paragraphs
F3061-17 Section 4.2.3 Applicable?
NO
YES
Minimize hazards for probable malfunctions
No process assurance objectives
Assign DAL per Table 2 or ARP4754A Section 5.2
ARP4754A Section 5.2
A- Catastrophic
B- Hazardous
C- Major
Table 2
All HW/SW at DAL C max except assessment level IV
SW Process Objectives Per DO-178
HW Process Objectives Per DO-254
Legend
Regulatory Layer
ASTM Guideline Layer
ASTM Guideline Contents Layer

Figure B-2 §23.2510 ASTM Safety Objective Development

ASTM F3230-17 identifies the use of Functional Hazard Assessments (FHAs) and System Safety Assessments (SSAs) methods described in ARP4761 [23] as acceptable means for establishing safety objectives and as a means for evaluating the final implementation to demonstrate satisfaction of these safety objectives. Based on F3230-17 failure condition severity classification, the quantitative objectives for a typical DEP using Assessment Level II would be per Table B-1.

In summary, the qualitative assurance objectives associated with DO-178C & DO-254 Level C and a quantitative assurance criteria of less than or equal to 10^{-7} failures per flight hour for catastrophic failure conditions would result from this methodology.

Table B-1 Quantitative FC Objectives per ASTM F3230-17

Assessment Level	Maximum Passenger Seating	Failure Condition Severity Classification			
		Minor	Major	Hazardous	Catastrophic
I	0 to 1 pax	≤ 10^{-3}	≤ 10^{-4}	≤ 10^{-5}	≤ 10^{-6}
II	2 to 6 pax	≤ 10^{-3}	≤ 10^{-5}	≤ 10^{-6}	≤ 10^{-7}
III	7 to 9 pax	≤ 10^{-3}	≤ 10^{-5}	≤ 10^{-7}	≤ 10^{-8}
IV	10 to 19 pax	≤ 10^{-3}	≤ 10^{-5}	≤ 10^{-7}	≤ 10^{-9}

B2.1.2 Approach 2- Using F3153-15 Standard for Objective Identification

The ability for an applicant to use system level verification as a means of a level of confidence in equipment hardware and software implementations is established in PS-AIR-23-09, System Level Verification of Electronic Equipment (Software and Airborne Electronic Hardware) for 14CFR Part 23 Airplanes [14]. This FAA Policy allows an Applicant of an airworthiness level I or II airplane to apply a process by which intended function and compliance with safety objectives of systems containing software and AEH may be verified by a combination of system level reviews, analysis and testing. This policy permits system level verification in lieu of software and AEH development assurance for systems functions which meet the Policy criteria.

Essential to the application of this Policy, is the need to capture the functional, operation, performance and safety requirements at an appropriate level of detail such the system requirements can be verified at the system level. The following objectives are required for compliance:

- System inputs and outputs are fully defined and documented,
- Defined, documented and verifiable intended functions and associated requirements at the system level and are testable and analyzable,
- Defined and documented failure conditions for the systems,
- Defined foreseeable operating conditions – including environmental conditions and the pass/fail criteria for qualification that must be shown to be satisfied individually and collectively,
- Defined and document anticipated abnormal inputs, behaviors and operation conditions,
- Defined and documented pass/fail criteria for evaluating the intended system functions,

- Document configuration management process, and
- A problem reporting process to capture, document and track issues to their resolution and closure.

A complete safety assessment (e.g. FHA, SSA, FTA, FMEA) is required by the Policy.

The objectives summarized in the bullet list above correlate to similar objectives captured in the ARP4754A [22] consensus standard objectives summarized in Table B-2.

Table B-2 Policy to ARP4754A Objective Correlation

Objective Area	Objective Numbers
Requirements Capture	2.3, 2.4, 2.5 & 2.7
Requirements Validation	4.1
Implementation Verification	5.1 thru 5.6
Safety Assessment	3.1, 3.4, 3.6 & 3.7
Configuration Management	6.1 thru 6.3

The FAA Policy identifies ASTM F3153-15 [18] as an acceptable guideline to accomplish the policy objectives. Figure B-3 highlights this alternate approach and its implications that software and airborne hardware will be verified through a system level process of error and failure identification rather than the DO guidance material.

In summary, the qualitative assurance objectives associated with ARP4754A FDAL A and a quantitative assurance criteria of less than or equal to 10^{-7} failures per flight hour for catastrophic failure conditions would result from this methodology.

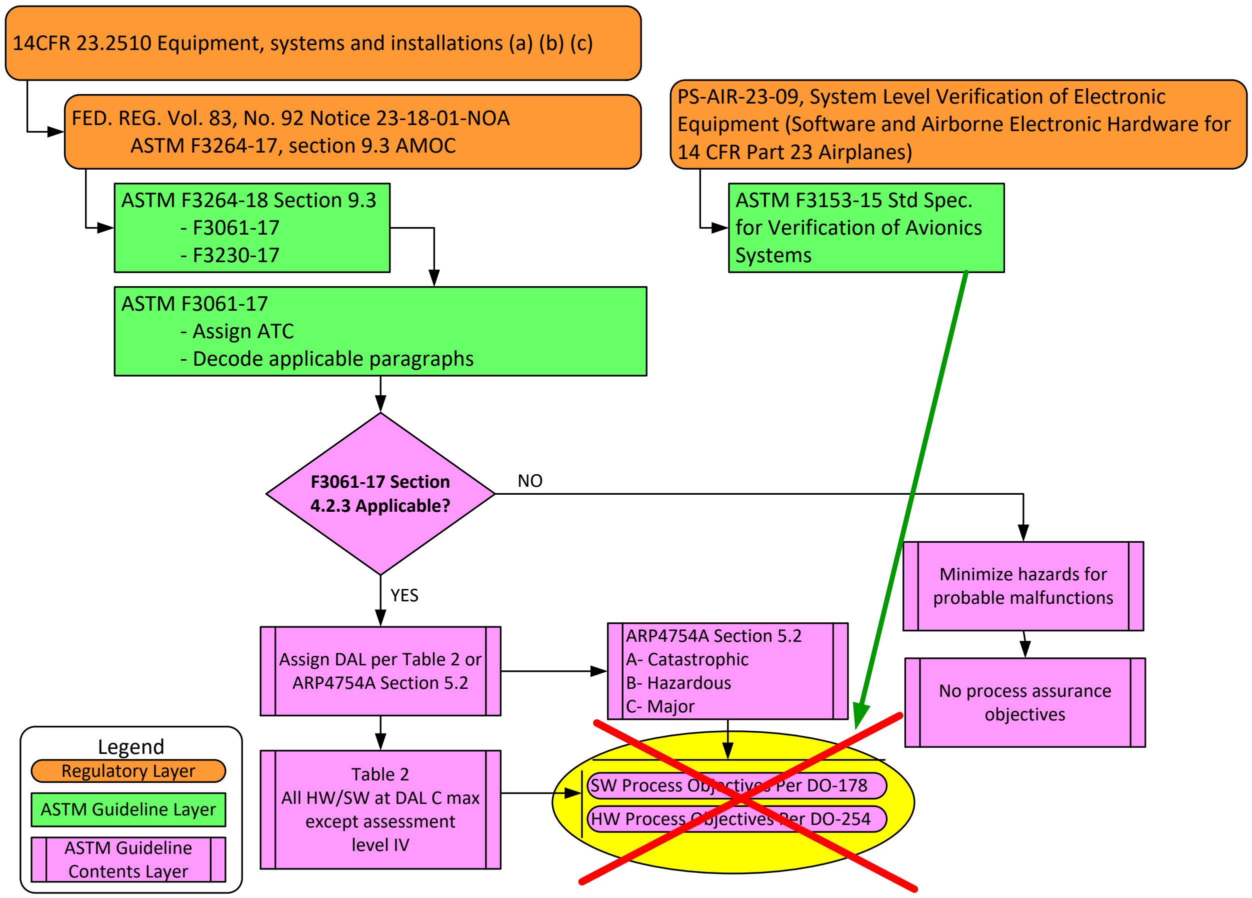

14CFR 23.2510 Equipment, systems and installations (a) (b) (c)
FED. REG. Vol. 83, No. 92 Notice 23-18-01-NOA
ASTM F3264-17, section 9.3 AMOC
PS-AIR-23-09, System Level Verification of Electronic Equipment (Software and Airborne Electronic Hardware for 14 CFR Part 23 Airplanes)
ASTM F3264-18 Section 9.3
- F3061-17
- F3230-17
ASTM F3153-15 Std Spec. for Verification of Avionics Systems
ASTM F3061-17
- Assign ATC
- Decode applicable paragraphs
F3061-17 Section 4.2.3 Applicable?
NO
YES
Minimize hazards for probable malfunctions
No process assurance objectives
Assign DAL per Table 2 or ARP4754A Section 5.2
ARP4754A Section 5.2
A- Catastrophic
B- Hazardous
C- Major
Table 2
All HW/SW at DAL C max except assessment level IV
~~SW Process Objectives Per DO-178~~
~~HW Process Objectives Per DO-254~~
Legend
Regulatory Layer
ASTM Guideline Layer
ASTM Guideline Contents Layer

Figure B-3 System Verification in Lieu of Implementation Processes

B2.2 Normal Category Rotorcraft

The FAA Normal Category Rotorcraft certification standards are defined in 14CFR Part 27 [4]. Rotorcraft are defined as “Heavier-than-air aircraft that depend principally for its support in flight on the lift generated by one or more rotors”. Normal Category Rotorcraft maximum weight must be less than or equal to 7,000 lbs (3182 kg).

Safety criteria for systems and equipment are defined in §27.1309 with AMOC defined in AC 27-1B, Change 8 [5] and PS-ASW-27-15, Safety Continuum for Part 27 Normal Category Rotorcraft Systems and Equipment [9]. These two documents recognize SAE ARP4754A [22] and SAE ARP4761 [23] as AMOC for rotorcraft functions and systems. They recognize DO-178C [25] and DO-254 [26] as AMOC for airborne software and airborne electronics hardware, respectively.

The advisory guidance is based on a risk versus assurance rigor approach using engine, passenger count and gross weight as the assessment level assignment criteria. Table B-3 summarizes the assessment level assignments based on engine and passenger count. Both quantitative and qualitative assurance criteria are defined.

As an example, a rotorcraft with a single turbine engine, carrying up to 4 passengers, weighing less than 4000 pounds is identified as Assessment Level II. Qualitative assurance criteria for Level II are set at “C” (both ARP4754A FDAL and DO IDALs) for functions and systems exhibiting catastrophic failure conditions. The catastrophic quantitative criterion is set to 10^{-7} failures per flight hour.

For a multi-engine rotorcraft, the FDAL and IDALs would be level “A” for the development processes with a quantitative assurance criteria of less than or equal to 10^{-9} failures per flight hour for catastrophic failure conditions.

Table B-3 Quantitative & Qualitative FC Objectives for §27.1309

Assessment Level	Engine & Max Passenger Seating	Failure Condition Severity Classification			
		Minor	Major	Hazardous	Catastrophic
I	Reciprocating 0 to 4 pax	$\leq 10^{-3}$ D	$\leq 10^{-4}$ C	$\leq 10^{-5}$ C	$\leq 10^{-6}$ C
II	1 Turbine 0 to 4 pax $\leq$ 4000 lbs GW	$\leq 10^{-3}$ D	$\leq 10^{-5}$ C	$\leq 10^{-6}$ C	$\leq 10^{-7}$ C
III	1 Turbine $\geq$ 6 pax 4000 $\leq$ 7000 lbs GW	$\leq 10^{-3}$ D	$\leq 10^{-5}$ C	$\leq 10^{-7}$ C	$\leq 10^{-8}$ B
IV	Twin Turbine	$\leq 10^{-3}$ D	$\leq 10^{-5}$ C	$\leq 10^{-7}$ B	$\leq 10^{-9}$ A

B2.3 EASA Special Condition for Small-Category VTOL Aircraft

Vehicles which combine a vertical takeoff and landing capability create a new category of air vehicle. These vehicles may or may not resemble conventional helicopters or airplanes. Their capabilities overlap criteria from multiple certification categories while containing implementation characteristics consistent with none of the categories. The integration of powerplants as part of the flight control function capability will require the engine to be certificated as part of the integrated airframe as opposed to separately from the airframe as in current practice. And these combined category vehicles may fit the definitions of both *powered-lift* and *rotorcraft*:

> Powered-Lift – “Heavier than air aircraft capable of vertical takeoff, vertical landing and low speed flight that depends primarily on engine-driven lift devices or engine thrust for lift during these flight regimes and on non-rotating airfoils for lift during horizontal flight” [1].
>
> Rotorcraft – “Heavier than air aircraft that depends principally for its support in flight on the lift generated by one or more rotors” [1].

The European Aircraft Safety Agency (EASA) has issued a VTOL Special Condition (SC) [11] of certification criteria for VTOL type vehicles under their jurisdiction. This document is targeted to VTOL aircraft of fewer than or equal to 9 passengers and a maximum take-off gross weight of ≤ 2,000 kg (4,400 lbs). EASA's SC was sent out for public comment on November 15, 2018 and was published on July 2, 2019.

EASA envisions two regulatory classes; Basic or Enhanced. The Basic Category is capable of controlled emergency landing after critical malfunction of thrust/lift. The Enhanced Category includes the Basic Category capabilities and in addition is planned for operations over congested areas and will be used for hire.

The initial AMOC for the VTOL SC will be either issued under EASA Notice of Proposed Amendment (NPA) or specific applicant project level consensus standards negotiated and accepted by EASA. The safety objectives were defined in SC VTOL.2510 Equipment, systems & installations [11]. EASA did not identify specific advisory material for the SC except a classification table was inserted in the special condition evaluation material (see Table B-4).

In summary, the qualitative assurance objectives associated would be FDAL/IDAL A and a quantitative assurance criteria of less than or equal to 10^{-9} failures per flight hour for catastrophic failure conditions would result from this methodology.

Table B-4 Qualitative & Quantitative FC Objectives for VTOL.2510

Category	Maximum Passenger Seating	Failure Condition Severity Classification			
		Minor	Major	Hazardous	Catastrophic
Enhanced	-	$\leq 10^{-3}$ FDAL D	$\leq 10^{-5}$ FDAL C	$\leq 10^{-7}$ FDAL B	$\leq 10^{-9}$ FDAL A
Basic	7 to 9 pax	$\leq 10^{-3}$ FDAL D	$\leq 10^{-5}$ FDAL C	$\leq 10^{-7}$ FDAL B	$\leq 10^{-9}$ FDAL A
	2 to 6 pax (Note A)	$\leq 10^{-3}$ FDAL D	$\leq 10^{-5}$ FDAL C	$\leq 10^{-7}$ FDAL C	$\leq 10^{-8}$ FDAL B
	0 to 1 pax (Note A)	$\leq 10^{-3}$ FDAL D	$\leq 10^{-5}$ FDAL D	$\leq 10^{-6}$ FDAL C	$\leq 10^{-7}$ FDAL C

Table B-4 Quantitative safety objectives are expressed per flight hour.

Table B-4 Note A: No considerations of system architecture for DAL reduction are acceptable.

B2.4 Current Practice Assurance Objectives Baseline

The current practice assurance baselines need to be extracted from the processes described in sections B2.1 thru B2.3. In the process of establishing the current industry assurance practice objectives it was discovered that the current practices do not have identified objectives. Current industry practices define activities that must be accomplished as the objectives. The “How” is defined but the underlying objective of “what” the activities are achieving is not defined. This was highlighted when the “ARP” and “DO” practices were compared against the ongoing NASA/FAA Overarching Properties Working Group efforts.

The underlying objectives of current practice activities must be extracted so that the project comparison task is comparing objectives to objectives rather than objectives to task activities.

The quantitative assurance “objectives”, across the different vehicle types evaluated in sections B2.1 thru B2.3 are well defined with universally accepted safety assessment methodologies identified to establish compliance.

The qualitative assurance objectives however define what to do rather than the goal or objective that the activities are trying to accomplish. This issue was also identified as an industry topic in [28] where the authors noted the objectives in DO-178C and ARP4754A *“might be more accurately described as techniques rather than objectives.”*

An internet search was initiated to see if extrapolation of current industry practice to level-of-confidence objectives had already been contemplated but nothing was found.

Table B-6 summarizes potential level-of-confidence (LOC) objectives associated with the current industry practices, sorted by failure condition severity classification. This LOC is a qualitative measure that the system and equipment performs its intended function and is safe.

No Effect – Functions that have been evaluated as having no airplane or system level failure effects have no level of confidence objective for the safe realization of the function.

Minor - Functions that have minor severity failures can have a low level of confidence of providing intended function or of being safe. The airplane level effects are limited so the qualitative and qualitative criteria should be minimal. {It is therefore surprising that current qualitative development criteria require such extensive process characteristics.}

Major - Functions that have major severity failures can have a moderate level of confidence of providing intended function or of being safe. Again for this failure condition the airplane level effects are limited so the qualitative and qualitative criteria should be minimal. {It is therefore surprising that current qualitative development criteria require such extensive process characteristics.}

Hazardous - Functions that have hazardous severity failures should have a high level of confidence of providing intended function or of being safe. A hazardous level function should have comprehensive set of development activities that link the aircraft level functional definitions to the low level design with evaluations of design decisions for safety impact.

Catastrophic - Functions that have major severity failures should have a very high level of confidence of providing intended function or of being safe. Catastrophic level functions should have the most comprehensive set of development activities that link the aircraft level functional definitions to all of the low level implementations.

The current industry practices result in the qualitative and quantitative objectives for catastrophic failure conditions summarized in Table B-5. These characteristics will be used in the alternate assurance concept comparison study.

Table B-5 Current Industry Assurance Baselines

Regulatory Part	Assigned Assurance Level	Level of Confidence and quantitative criteria
14CFR Part 23	C	Moderate level of confidence characteristics and LTE 10^{-7} fph
14CFR Part 27	C	Moderate level of confidence characteristics and LTE 10^{-7} fph
	A	Very High level of confidence characteristics and LTE 10^{-9} fph
EASA VTOL SC	A	Very High level of confidence characteristics and LTE 10^{-9} fph

Table B-6 Perceived LOC to Equipment Characteristics Correlated to Failure Condition

Failure Condition Severity Classification	Failure Condition Effects for Airplane, Occupants, Crew	LOC	Current Industry Equipment Development Process Characteristics	DAL Level
No Effect	-	-	-	E
Minor	A/C: Slight reduction in functional capability or safety margins OC: Physical discomfort CR: Slight increase in workload or use of emergency procedures	Low	• Implementation is developed per a described and independently monitored process. • Implementation is verified to provide captured intended functions and behaviors with partial linkage of required airplane/system definitions to high level implementation definitions. • Implementation may contain unverified and unlinked implementation characteristics.	D
Major	A/C: Slight reduction in functional capability or safety margins OC: Physical distress possibly including injuries CR: Physical discomfort or significant increase in workload	Moderate	All Low LOC Characteristics Plus: • Implementation is verified to provide the captured and validated intended functions and behaviors with linkage from required airplane/system definitions to lowest level required implementation definitions. • High level design decisions have been evaluated against safety objectives.	C
Hazardous	A/C: Large reduction in functional capability or safety margins OC : Serious or fatal injury to an occupant CR: Physical distress or excessive workload which impairs ability to perform tasks	High	All Moderate LOC Characteristics PLUS • Low level design decisions have been evaluated against safety objectives.	B
Catastrophic	A/C: Loss of aircraft OC: Multiple fatalities CR: Fatal injury or capacitation	Very High	All High LOC Characteristics Plus: • Implementation does not contain unverified or unlinked implementation characteristics.	A
		Extremely High	All Very High Characteristics Plus: • Implementation includes architectural and/or diversity mitigation techniques.	

Note: A/C – Aircraft; OC – Occupants; CR – Flight Crew

Appendix C: Run-Time Assurance Background

C.1 Run-Time Assurance Techniques

Run-time assurance (RTA) is an algorithmic architecture that has been suggested as a technique for designing systems that must satisfy competing sets of requirements. Examples include:

- systems that deliver highly tuned performance while being robust to physical variations;
- high-assurance systems that are capable of autonomous operation; and
- systems that operate reliably while allowing frequent updates to add new features.

These architectures satisfy competing requirements by incorporating multiple distinct algorithms, each of which achieves a subset of the requirements, and dynamically switching between them at run-time to ensure that all requirements are met. Note that while the literature on RTA architectures is primarily focused at the algorithmic level, the concepts involved could also be applied to physical components or to combined hardware/software subsystems.

An example of an RTA-based control architecture is illustrated in Figure C-1. This design pairs a main control algorithm with a backup algorithm and a monitoring/switching component. At regular time intervals, the RTA Monitor & Switch dynamically selects one of these two algorithms to control the plant over the duration of the next interval. Its decision is based on a real-time analysis of the original design requirements and may account for the current state (and state history) of the plant, the operating environment, and commanded inputs.

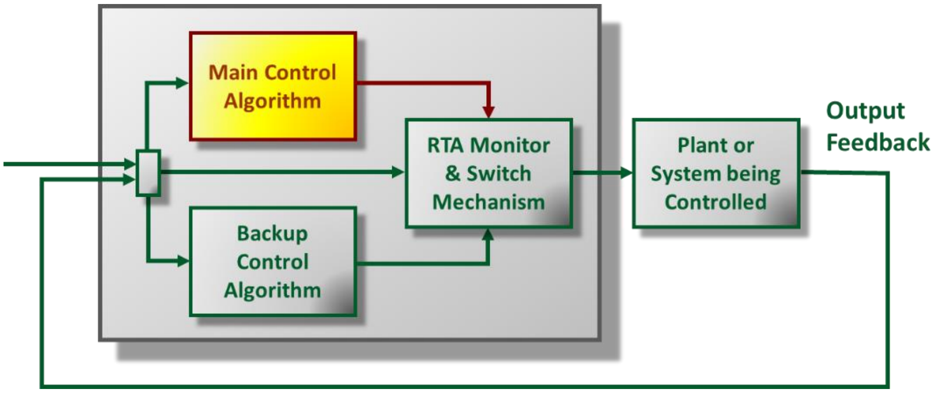

Main Control Algorithm
RTA Monitor & Switch Mechanism
Backup Control Algorithm
Plant or System being Controlled
Output Feedback

Figure C-1 High-Level Diagram of an RTA-Based Control System

RTA architectures like that above are often suggested as a technique for increasing confidence in autonomous or adaptive control systems. In the published literature there appear two different approaches for assigning roles to the main and backup control algorithms, which we will refer to as *Main Adaptive* and *Backup Adaptive* architectures:

Main-Adaptive architectures restrict autonomous/adaptive algorithms to the main controller and use traditional control design for the backup.

Backup-Adaptive architectures use a traditional control algorithm as the main controller, and place damage-adaptive capabilities in the backup.

Main-adaptive RTA architectures typically pair an advanced (and to some degree untrusted) main control algorithm with a traditional (and high-assurance) backup controller. The RTA monitor is designed to switch to the backup controller if the vehicle is otherwise healthy but is approaching an unsafe state. Since the reversionary controller is typically a simpler design, it will not have the performance or robustness capabilities that the advanced system possesses but is designed to have the minimum required capabilities to recover safe operation. In this manner, RTA architectures bound the behavior of an untrusted advanced system, allowing it to operate and provide the benefits of its advanced capabilities, but disallowing any unforeseen, unsafe actions that could compromise system safety. This arrangement is intended to guard against errors in the design or implementation of the main controller.

Backup-adaptive RTA architectures typically use a main controller that is highly trusted under the assumption that the plant being controlled is performing nominally. The backup controller is intended for use only in the case that the plant behaves in a significantly off-nominal way or the operating conditions are far from those considered at design-time. The backup controller may or may not be highly trusted, depending on the specific application. The RTA monitor switches to the backup controller if it appears that the plant dynamics or operating conditions are such that using the main controller is likely to result in an unstable or otherwise unsafe condition. In cases where the backup controller is not highly trusted, this is a last-ditch effort to prevent a catastrophic failure of the system. In both cases (trusted and untrusted backup), this arrangement is intended to guard against physical failures involving the plant or accidental operation outside of design conditions.

Publications addressing RTA as a general design technique began appearing approximately 25 years ago from researchers at Carnegie Mellon University’s Software Engineering Institute, in the context of industrial control, and from researchers at Barron Associates, in the context of aircraft control. Since these initial publications, these and many other research groups have matured the concepts involved, performed case studies of the approach, and developed numerous supporting technologies. This section contains a broad-based review of these research efforts. This review addresses only what is available in the open literature; efforts not reflected in journal papers, conference publications, or open technical reports are not covered. We focus here only on RTA publications that are relevant from a control systems perspective. This review specifically excludes work focused primarily on detecting and responding to software security attacks. Similarly, we exclude work that focuses on system reset or reboot as the solution to a detected problem.

C1.1 Concepts for RTA-Based Control

This section reviews published research on various concepts for how RTA architectures can be used in control applications. The focus is on architecture concepts, not the tools for building implementations of those architectures.

C1.1.1 Simplex Architecture

The Simplex architecture, first explored by Lui Sha and colleagues at Carnegie Mellon University's Software Engineering Institute, is frequently cited within the RTA literature. Most of this work explores designs that are consistent with main-adaptive architectures, which use lesser-trusted algorithms in main controller and simpler, highly-trusted algorithms in the backup controller. The Simplex architecture was originally designed to support online upgrading of industrial control software while ensuring that the new software does not lead to an unstable (or unsafe) condition. In [29], the authors formulate the Simplex architecture in terms of *replacement units*, which are software implementations with standardized communication interfaces and are designed to facilitate replacement of one with another during run-time. They define three types of replacement units:

1. Application units implement the control functions of a subsystem.
2. Safety units implement a safety control algorithm and are used if the physical plant does not have fail-safe operation.
3. Management units facilitate the replacement of one application unit with another.

If a safety unit is part of the design, all communication from the application units passes through the safety unit, so it can exert control when and if it becomes necessary.

The application units are designed to have analytic redundancy, by which the authors mean that they are distinct software implementations that satisfy a set of common requirements. They contrast this with functional redundancy, referring to different software implementations of mathematically equivalent algorithms (similar to multi-version software discussed previously), and replication redundancy, referring to the same software implementation running on separate hardware. Two subsystems with either functional or replication redundancy will produce the same outputs when given identical input sequences.

A transition process is used to replace one application unit with another while the system remains operational. The new unit is created, and both input and plant state information are synchronized between the new and old units. The new unit is allowed to execute until its behavior converges, at which point its outputs are substituted for the old unit, and the old unit is removed.

The authors in [30] describe the Simplex Architecture as a middleware technology and examine its use in motion control applications. Here, the Simplex architecture is specifically described in terms of an advanced, "yet-to-be-proven" controller and a separate safety controller. They discuss the use of monitoring software that analyzes system safety and performance, engaging the safety controller before a system failure occurs. They authors build a model of this Simplex design in Communicating Sequential Processes (CSP), a formal language for describing concurrent systems. They demonstrate use of the FDR model checker to prove properties about the architecture, such as absence of deadlock, liveness, and command sequencing.

Seto, in references [31] and [32], describe a variation of the Simplex Architecture specifically intended for single-processor implementations. The design is intended to protect against both timing faults, such as a failure to generate a control command within an acceptable period, and semantic faults, such as generating a control command that would violate specifications. As illustrated in Figure C-2, this design incorporates implementations of three separate control algorithms:

1. an experimental controller that may include new and/or untested features;
2. a baseline controller that has been extensively tested; and
3. a safety controller that is highly robust but not necessarily capable of satisfying all performance requirements for the combined system.

The experimental and baseline controllers are designed to be analytically redundant, in the sense previously described. However, the safety controller is different. It is designed to take over from the experimental controller, when necessary, and steer the plant to a state suitable for the baseline controller to begin operation. Thus, the safety controller is expected to be highly reliable and to have a very large domain within the state space, larger than that of the baseline controller. The safety controller is not intended for long-term use, however, and is not designed to meet the same performance requirements as the baseline or experimental controllers.

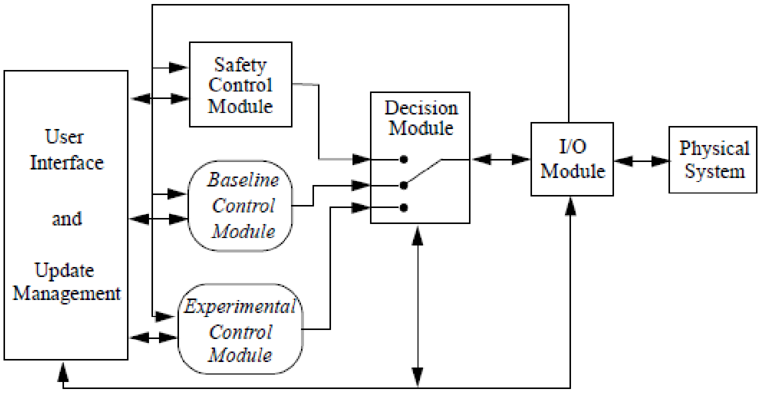


Figure C-2 Simplex Architecture [32]

They then consider possible state sequences for the decision module and the corresponding switching conditions when the plant is a continuous-time system whose state can be described by a differential equation. To this end, they define the operating region of the baseline controller to be the set of plant states from which the baseline controller will continue to satisfy all its performance and safety requirements. A corresponding operating region for the safety controller is defined as the set of plant states from which the safety controller can continue to meet all its safety requirements and can steer the plant into the baseline controller's operating region. The decision module's switching behavior can then be summarized as the following:

> Switch from the experimental controller to the safety controller if the plant state reaches the boundary of the safety controller's operating region. Subsequently, switch from the safety controller to the baseline controller when the plant state enters the baseline controller's operating region.

In reference [33], the authors define operating regions for the baseline and safety controllers to be the stability regions of the closed-loop system consisting of the plant and corresponding controller. Using results from Lyapunov theory, they demonstrate how to compute these operating regions for linear systems.

Finally, Crenshaw's work [34], generalizes the Simplex Architecture concept to arbitrary control applications and notions of safety. Assumptions that the plant being controlled follows linear continuous-time dynamics and that safety is equivalent to stability are dropped. The authors present a reference model for this generalized Simplex architecture, and they develop formal logical statements of system safety and recoverability in terms of this reference model.

C1.1.2 Simplex Architectures for Embedded Systems

The Simplex Architecture discussed in the previous section is focused primarily on providing tolerance to software faults. As such, its implementations often entail multiple controllers and the RTA monitoring components running as separate processes on the same processor. This arrangement does not provide tolerance to faults in the operating system or middleware, nor does it provide tolerance to faults in the computational hardware itself. In Bak [35], the authors describe the System-Level Simplex architecture, which partitions these algorithmic components onto separate hardware. They present an AADL model specification for the architecture and describe model checking tools to ensure that designs based on the specification satisfy a number of critical properties. They also describe a tool to generate VHDL code for the safety controller and monitor, the output of which is suitable for constructing ASIC or FPGA implementations of those components. The System-Level Simplex Architecture is demonstrated as a technique for fault tolerant unmanned air vehicle control in Vivekanandan [36].

C1.1.3 Network-Centric Simplex

If the components of a Simplex architecture are connected through digital communication networks, the delay or dropping of messages may compromise the design's effectiveness. In Bak [37], the authors consider a scenario in which a high-level (supervisory) control algorithm directs the operation of one or more controlled subsystems through a network that may silently fail to deliver commands. They describe the logic for a run-time assurance component that prevents a command from being sent if the possibility of a future communication failure might render those commands unsafe. They prove safety and progress properties for this design.

In separate work, the NetSimplex architecture is introduced in Yao [38], which addresses problems of bounded network delay between components in a standard Simplex design. The authors assume that the backup controller is a traditional gain-scheduled controller designed around a linear system model. They allow for a single uncertainty value within the plant and assume that the plant dynamics change as a linear function of this uncertainty. They address how to maximize the stability region of the backup controller relative to the range of this uncertainty, while minimizing the reduction of that region due to communication delays.

C1.1.4 Robust Simplex

Variations of the Simplex Architecture discussed to this point are designed to provide tolerance to faults in the control software, control hardware, and communication network. In Wang [39] [40], the authors develop Simplex variations that can provide tolerance to faults in the plant being controlled. These architectures have an adaptive controller as their backup, which is designed to be robust with respect to variations in the plant dynamics. The main controller is assumed to be designed for high-performance and might not be fully trusted. The backup controller is enabled if one of two conditions occurs:

1. The system model around which the main controller was developed appears to be significantly different than the plant currently being controlled; or
2. The plant state approaches the border of a pre-defined switching region.

An occurrence of the first condition would suggest a potential plant failure, in which case the adaptive controller would generally be preferred to the high-performance main controller. An occurrence of the second condition would suggest a fault in the main controller. Wang [39] is a preliminary version of this work, which assumes that the plant is a linear system, even in the presence of physical failures. Its backup controller is an L1 Adaptive controller. Wang [40] assumes that the plant is a linear system when functioning correctly but may become nonlinear in the presence of a physical failure.

It is worth noting that the Robust Simplex Architectures have properties that are similar to both main-adaptive and backup-adaptive architectures, as described in the introduction of Section C.1. That is, like the main-adaptive architectures, it is allowed that the main controller may be untrusted but that the L1 control algorithm implemented in the backup controller can be assured to a high level. However, like the backup-adaptive architectures this L1 controller designed to adapt to physical variations or failures in the plant. Assurance of the L1 controller is revisited in Section C1.13.

C1.2 Multi-Level Interacting RTA

Researchers at Barron Associates were the first to develop RTA as an architecture for ensuring the safety of aircraft systems, exploring applications to inner-loop aircraft control, guidance and collision avoidance, flight and mission management, turbofan engine control, and UAV geofencing.

The notion of RTA specifically as a technique to facilitate certification of autonomous or adaptive control algorithms for aircraft is introduced in Bateman [41]. This paper introduces the main-adaptive formulation of RTA, placing adaptive capabilities in the main controller and using a traditional backup controller. The architecture assumes that the RTA monitor bases its switching decision, in part, on information from a fault-detection system that can reliably detect off-nominal aircraft dynamics. If the vehicle dynamics are nominal and a fault in the main controller is detected, the RTA monitor will switch to the backup (or failsafe) controller. However, if the vehicle itself is exhibiting some failure condition, the RTA monitor will allow the main controller significant freedom to adapt, in an effort to spare the aircraft. These combinations are illustrated in Figure C-3.

The authors visualize the RTA architecture as a safety wrapper ensuring that untrusted software systems inside the wrapper do not lead to hazardous conditions. The wrapper and its contents become a trusted software module that can be used in conjunction with other, similarly wrapped software components. They envision a fine-grained composition of such components in such a way that would allow graceful degradation of capabilities by selectively disabling individual malfunctioning components, while allowing other advanced components to continue operating. This concept is illustrated in Figure C-4.

	Vehicle Nominal	Vehicle Failed
Controller Nominal	Controller obeys tight nominal bounds	Controller allowed significant freedom to adapt
Controller Failed	Failsafe controller recovers safe behavior	*Recovery of safe flight may be impossible* Extremely low probability scenario

Figure C-3 Failure Combinations for the Vehicle and Adaptive Main Controller [41]

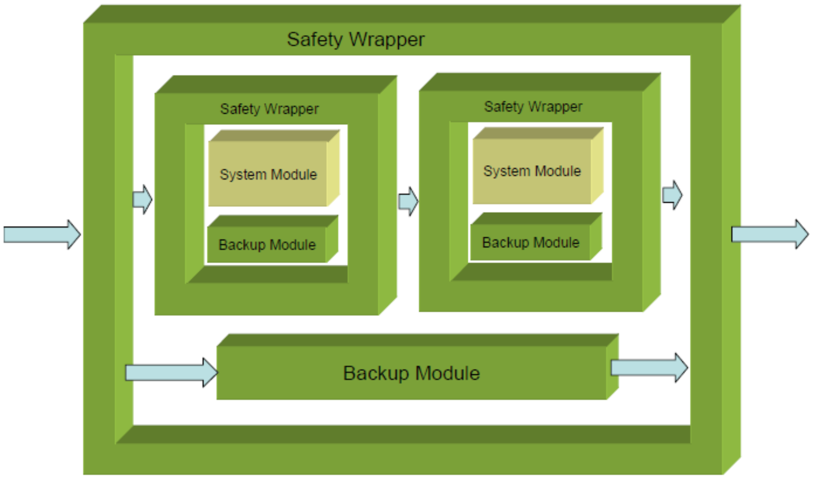


Figure C-4 A System with Multiple Interconnected RTA-Protected Subsystems [41]

The concept of an RTA Manager component to coordinate among individual RTA monitors is introduced in Schierman [42]. The idea is that for designs with multiple RTA-protected subsystems, the switching of one subsystem to its backup system may necessitate the switching of other subsystems. The authors demonstrate the use of multiple nested RTA-protected subsystems on a high-fidelity hardware-in-the-loop simulation of Lockheed Martin's Sea Based Endurance UAV. Main controllers, backup controllers, and RTA monitors were constructed for the:

1. outer-loop guidance subsystem,
2. inner-loop control subsystem,
3. desired dynamics / flying qualities subsystem, and
4. effector blender / control allocation subsystem.

The RTA manager was designed to ensure that, if a fault were detected by multiple RTA monitors at different levels within a subsystem hierarchy, the inner-most RTA monitor would switch to its backup controller first. If RTA monitors at higher levels in the hierarchy continued to indicate a fault, the next lowest level would switch, etc.

This notion that RTA architectures may be simultaneously used at multiple levels of a control hierarchy is expanded in Schierman [44] to address nested feedback control architectures spanning the entire range from inner-loop control of an individual vehicle up through command and control of vehicle fleet.

As illustrated in Figure C-5, the multi-level interacting RTA architecture includes:

1. Fleet-level mission planning systems (MPS),
2. Vehicle-level flight management systems (FMS),
3. Outer-loop guidance algorithms (GLAW),
4. Collision avoidance algorithms (CAS), and
5. Inner-loop control algorithms (CLAW).

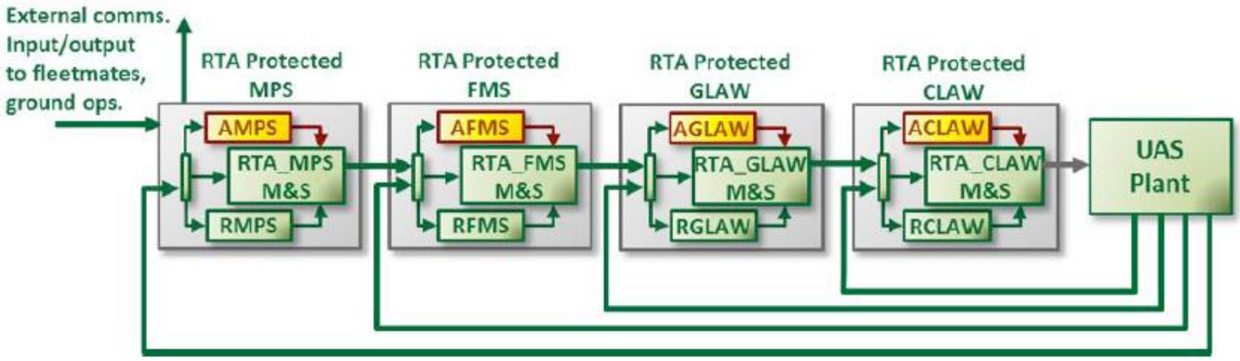


Figure C-5 Multi-level Interacting RTA Architecture [44]

The authors note that, from the perspective of each control level, the assemblage of all downstream components constitutes a plant under control. For example, from the guidance law perspective, it is controlling a plant composed of the inner-loop controller and physical aircraft. From the flight-management system's perspective, it is controlling a plant composed of the guidance law, inner-loop controller, and physical aircraft. This has several consequences. First, the notion of system state is different for each controller, as is the notion of system safety. For example, while the inner-loop controller is concerned with issues like structural and attitude limitations, the guidance law is concerned with issues like trajectory errors and separation. Second, what constitutes a performance requirement at one level may be a safety-related issue at a higher level. For example, if the inner-loop controller is unable to satisfy required transient response characteristics, the guidance law may be unable to ensure separation. The authors enumerate the categories of behavior checks that must be performed by each RTA monitor as a result of these interactions.

Finally, the authors formalize a general concept of safety, as well as the switching boundaries deriving from it, that can be applied in all levels of the multi-level RTA architecture. Their approach is defined in terms of the state space being managed by a controller and the subset of that space that is known (or defined) to be safe, from that controller's perspective. It explicitly accounts for the amount of time it takes to transition to the backup controller, any transients associated with that transition, defined procedures to be followed in the aftermath of such a transition, and the timing intervals between successive checks by the RTA monitor. The approach defines three nested subsets of the state space, as illustrated in Figure C-6. The Type I safety region is that subset in which the plant being controlled is known to be safe; ensuring that the plant stays in this region of the state space is a primary requirement of the RTA architecture. The Type II safety region is a subset of the Type I region, such that the backup controller can execute a pre-defined recovery operation and any transients associated with switching to the backup controller do not cause the plant to exit the Type I region. Finally, the Type III safety region is a subset of the Type II region, such that the main controller cannot steer the plant outside of the Type II region between two consecutive checks by the RTA monitor. The RTA monitor must be designed to transition to the backup controller if the plant state is ever found to be outside the Type III region.

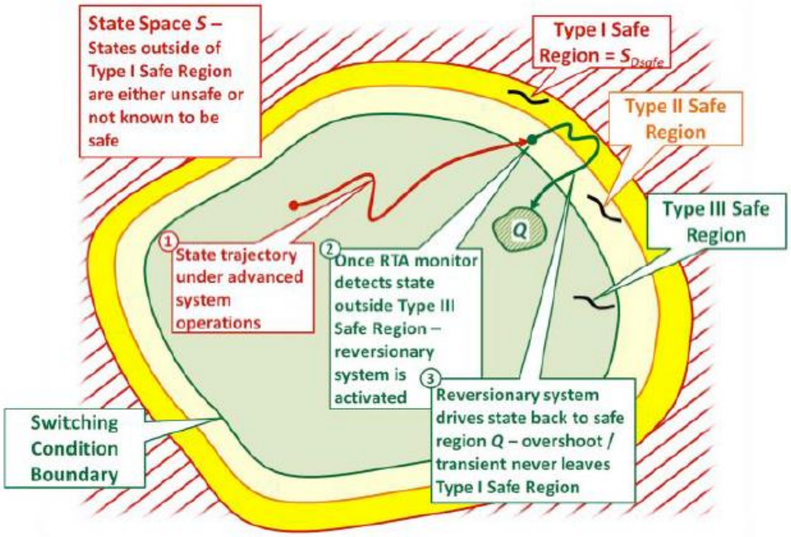

State Space S – States outside of Type I Safe Region are either unsafe or not known to be safe
Type I Safe Region = S_{Dsafe}
Type II Safe Region
Type III Safe Region
Q
① State trajectory under advanced system operations
② Once RTA monitor detects state outside Type III Safe Region – reversionary system is activated
③ Reversionary system drives state back to safe region Q – overshoot / transient never leaves Type I Safe Region
Switching Condition Boundary

Figure C-6 Illustration of Type I, II, and III Safety Regions [44]

A significantly expanded version of this work is presented in Schierman [43]. There, the authors examine the issues surrounding the potential decrease in performance that could occur when an RTA monitor commands a transition from its main to its backup controller. Since performance at one level can impact safety at a higher level, the fact that a down-stream component has transitioned to its backup controller must be communicated upstream, where appropriate actions may need to take place. For example, if the inner-loop subsystem delivers slower rise or settling times after switching to its backup controller, the guidance system may insist on greater separations, which in turn may cause the flight-management system to generate alternate paths with wider turning radii and wider tracking tolerances. The authors define a dynamic contract-based approach that leverages formal assume-guarantee compositional reasoning techniques.

The authors present a detailed example of the multi-level interacting RTA architecture, incorporating the components shown in Figure C-5. The example defines inputs and outputs for each level, describes candidate algorithms for the main and backup controllers at each level, applies the safety formulation to each level and derives the resulting switching algorithm, and defines the sets of performance contracts that can be offered by each level.

C1.3 Manual Pilot Recovery RTA

Expanding on the Barron Associates RTA formulation, researchers at the University of Tulsa, AFRL, NASA, and the FAA examined the use of RTA to enable low-cost retrofit of autopilot systems to GA aircraft. In their design, the autopilot fills the role of main controller, and a human pilot fills the role of backup. The RTA monitor checks that the aircraft remains within a predefined region of the state space, forcing control to the pilot if the autopilot exits that region.

In Hook [45], the authors formulate a high-level assurance case structure that argues for the safety of the combined system and discuss the evidence that might be generated to support each claim in that argument. The claims (or goals) of this assurance case are roughly equivalent to a set of high-level safety requirements for the combined system. In Fuller [46], the authors describe a hybrid model of the combined autopilot/RTA-monitor/human-pilot system that accounts for pilot delay upon switching and other factors. They use this model to design boundary regions for the RTA monitor. In Fuller [47], this arrangement is expanded for remotely-piloted aircraft, adding a second level of RTA components in which the pilot is viewed as an advanced controller, and an automated recovery controller is provided to deal with potential disasters arising from (the likely inevitable) automation surprise. Conceptually, this design is similar to the Simplex arrangement in Figure C-2, where the human pilot fills the role of baseline controller and the second-level automatic recovery controller fills the role of safety controller.

C1.4 Multi-Monitor RTA

The Multi-Monitor RTA proposed in Hook [48] envisions a fine-grained partitioning of RTA-protected subsystems at the function level, in a manner that is similar to the multi-level interacting RTA described in Section C1.2. Developed as part of the NASA/FAA Traveler Project, this research takes a “flight-functional” approach to gradually increase automation. The idea is that, rather than rolling out low-levels of automation across multiple functions, one-by-one individual functions are fully automated to a high enough degree that the pilot is not required to oversee them. Those functions would be partitioned to form a set of highly modular, largely non-overlapping tasks. They point out that this could allow each algorithm to be somewhat simpler in construction, and thus more easily developed and tested. Additionally, as each function is automated, it is encapsulated with its own RTA monitor and backup system.

According to Skoog [49], this project ongoing and is being conducted in three phases. In the first phase, requirements for coordination of run-time monitors and interfaces are being developed. Identifying sensor and monitor requirements is an activity to be performed in the second phase, which also addresses the case for airworthiness case. Third phase applies the Multi-Monitor RTA concept to several aircraft and concepts of operation.

According to MTSI [50] and Skoog [51], the second phase of this project was started in October 2018, under the Resilient Autonomy project, a joint capability technology demonstration (JCTD) funded by the OSD. It seeks to build a demonstration on the HQ-90 hybrid quadcopter. RTA monitors will include geofencing, ACAS, GCAS, separation, weather avoidance, and person avoidance.

C1.5 Justified Speculative Control

The ModelPlex technique, introduced in Mitsch [52], couples design-time proofs that a control algorithm satisfies a set of safety and/or performance requirements with run-time checking that the plant being controlled satisfies the dynamics model assumed during design time. The concept is that if the plant's dynamics model is found to differ from what was expected, then the controller design may no longer meet its requirements, and a backup controller can be engaged. Platzer [53] gives a high-level discussion of the value of the ModelPlex approach, focusing on the differences between the behavior of an idealized model and that of actual systems.

This ModelPlex concept is used in Fulton [54] as part of an RTA architecture the authors term Justified Speculative Control. In this approach, an adaptive control algorithm is used as the main controller, but its adaptation is limited to a range that can be proven safe at design-time under assumptions about the plant dynamics. Those assumptions are checked by the run-time monitor. If it finds that the assumptions are violated (i.e., the operational system is not consistent with the assumed plant dynamics), the restrictions on the adaptive controller are removed. The intent is to provide the best chance of salvaging the system when it has become physically damaged or is operating in unintended environmental conditions.

In this design, there is only a single controller; however its allowable range of adaptation is determined by the RTA monitor. Logically, this is similar to the robust simplex techniques of Section 0, except that in this case both the main and backup controllers are adaptive. Because it uses a trusted algorithm as the main controller and an untrusted adaptive controller as the backup, this design is consistent with the backup-adaptive RTA architectures.

C1.6 RTA Case Studies

This section discusses a number of recent publications that look at specific aerospace applications of RTA architectures rather than on the development of general RTA concepts.

The use of formal methods analysis to prove safety properties of an RTA architecture is explored in Gross [55]. The authors focus on a system for attitude control of a 6-U CubeSat satellite through a reaction wheel array. The backup controller implements a rate-limited PID control algorithm, and the RTA monitor checks each command to see whether it violates bounds on maximum angular acceleration an whether it would lead to an over speed condition. They use a combination of formal methods tools to prove that this combination will ensure that the reaction wheel array will not be driven outside its operational bounds, regardless of what any main controller might command.

In Avram [56], the authors consider the use of RTA techniques for adaptive control of a quadrotor. An adaptive main controller is used for robustness to modeling uncertainties and to accommodate potential physical faults that could cause partial loss of effectiveness in the rotors. An RTA architecture is used to provide tolerance to software faults in this adaptive controller. Its backup controller is a traditional linear control algorithm with fixed gains that is intended to stabilize and land the quadrotor in case the main controller misbehaves. The RTA monitor switches to the backup controller if tracking errors under the main controller ever exceed an upper bound that is provable for its adaptive algorithm at design-time. They demonstrate this system using an actual quadrotor an indoor test environment, with a high-resolution position and attitude tracking system for data collection.

This architecture is investigated in Dillsaver [57] from the perspective of military airworthiness certification under MIL-HDBK-516C [123]. The authors look at the challenges to certification of the RTA monitor and switching components that are posed by various criteria from that document and discuss examples of simulation artifacts that could be generated to provide evidence of compliance.

The case study described in Phan [58] involves two RTA architectures used simultaneously at two different control levels of a simulated unmanned ground vehicle. One RTA-protected subsystem, termed the mission planning component, is designed to ensure that the vehicle returns to an available charging station before its stored energy is completely drained. Another, termed the navigation component, is designed to ensure that obstacles are avoided. The authors develop switching logic for the RTA monitors and formulate a pair of assume-guarantee contracts that facilitate proofs of system-level behaviors.

An application of RTA architectures to control of turbofan engines is presented in Schierman [59]. The authors design the RTA monitor to augment engine protection logic, which is commonly used on such engines to prevent the engine from exceeding its physical limits by regulating fuel flow rate. The main controller is a model-based engine control algorithm incorporating a self-tuning Kalman filter to provide high-precision control, and a traditional engine control algorithm is used as the backup. In this application, the RTA architecture is used to ensure a minimum set of performance requirements, while providing higher performance characteristics as long as the main controller functions correctly.

An application of the RTA architecture to geofencing of small UAVs operated in beyond-line-of-sight conditions is presented in Bateman [60]. In this design, an RTA architecture is used to ensure that arbitrary advanced path planning and guidance algorithms do not steer the aircraft outside of a pre-defined geographical area. The backup controller is designed to construct a Dubins path from the vehicle’s initial position to a tangent point intercepting a circular loiter pattern around a pre-selected safe location. The RTA monitor implements the Type III safety boundary from the multi-level interacting RTA architecture, which was discussed in Section C0.

C1.7 RTA Switching Conditions

One of the most difficult issues with the design and implementation of an RTA architecture is determining the switching conditions for its RTA monitor to ensure that the system never violates its safety requirements. Several recent publications have explored techniques to compute these conditions and to construct appropriate software implementations based on them.

In Rudd [61], the author demonstrates the use of real-time reachability software within an RTA monitor that switches between an advanced (main) trajectory generation algorithm and a trusted backup. The RTA monitor partitioned the space of possible 4-D location-time coordinates of a modeled aircraft into safe and unsafe combinations. If the main algorithm produced an intended trajectory such that a safe combination was not reachable by the aircraft, the backup system was used to plan a new trajectory. This formulation was demonstrated on examples of slalom maneuvers, glideslope reacquisition, and no-fly zone avoidance.

In Bak [62] [63] , the authors develop an approach for off-line computation of the switching boundary using back-reachability calculations. The back-reach of a given state is the set of states from which a trajectory could be constructed that passes through that given state. The authors explore algorithms for over-approximating the back-reach of states for plants that have hybrid dynamics. By applying their technique to each point on the boundary separating safe and unsafe states, they can compute a switching boundary that will ensure the RTA monitor enforces safety guarantees. They describe a toolkit for computing such a boundary and for generating source code for an RTA monitor that enforces this boundary.

In Bak [64] and Johnson [65], the authors point out that off-line approaches based on back-reach calculations can be overly conservative due to the over-approximation techniques employed. As an alternative, they propose the use of run-time reachability calculations for hybrid systems. In their formulation, the RTA monitor periodically performs a reachability calculation from the current state to determine if the plant can reach a non-recoverable state, which is any state from which the backup controller may steer the plant into an unsafe state. While forward reachability calculations also employ over-approximations, they tend to be much more conservative than for back-reach calculations. The authors demonstrate their approach running on embedded hardware with real-time performance guarantees.

A similar approach is taken in Yang [66], which describes the use of reachability calculations to determine whether the main controller can steer the plant to an unrecoverable state. However, they perform off-line computations to characterize the boundary between recoverable and unrecoverable states. Thus, their online reachability calculations can terminate if the reach set is found to intersect this boundary. This reduces the amount of online computations required and can reduce the amount of over-approximation involved. The boundary is represented in the form of a barrier certificate, a set of functions constructed such that their zero-level set is equivalent to the boundary.

Whereas the previous research assumes an essentially arbitrary main controller, Zhang [67] focuses on the design of an RTA monitor that is customized for use with an adaptive main controller implementing a neural network. The goal is to detect unstable learning behaviors due to either software coding errors or physical faults. The neural network controller is proven to stably adapt as long as several key assumptions (bounds on the approximation error, bounds on modeling error, and actuator saturation) are met. They derive the logic for an RTA monitor and prove its behavioral properties, in terms of both false alarms and the time to detect actual fault conditions. The system and controller are assumed to be nonlinear. They explicitly model the effects of both software errors and physical faults on the learning system.

C1.8 Programming Frameworks

In Desai [68] and [69], the authors describe efforts to develop high-level programming language support for the development of systems that involve the composition of multiple RTA-protected subsystems, such as the multi-level interacting RTA architectures of Section C1.2 and the multi-monitor RTA architectures of Section C1.4. The goal of this development is to allow developers to use declarative programming constructs to specify individual RTA modules, from which code will be automatically generated in the C programming language to implement the corresponding RTA monitor. The monitor is designed to enforce safety properties based on a decomposition of the state space that is similar to the Type I, II, and III safety regions discussed in Section C1.2. Construction of those regions is accomplished through the use of online reachability calculations.

C1.9 Run-Time Verification

Run-time verification is a very active research area within computer science and related communities. The focus in much of this work is on formally stating, at design time, a set of conditions that must hold true during run-time for a software system, then automatically generating monitors that can observe the executing code to detect any fault conditions. In general, the focus is primarily on detecting such faults. Details of what actions should be taken following fault detection are highly application specific, and they are often not directly addressed.

Nevertheless, there are significant similarities between the run-time verification and the run-time monitor at the heart of RTA architectures. Both continually monitor a system for the occurrence of a formally defined condition, then invoke a pre-specified action. Note that there is a slight distinction in the conditions these two technologies are designed to detect. Run-time verification typically detects a prohibited condition at the time it occurs, or immediately afterwards. In contrast, the RTA monitor must avoid the prohibited condition before it occurs, accounting for any momentum in the physical plant being controlled, integrator windup in switched control blocks, etc.

Many papers have been published in this area, going back several decades. The annual International Conference on Runtime Verification has run continuously since 2001, and links to proceedings from past conferences can be found here: https://www.runtime-verification.org/. Rather than attempt to survey this very rich field, we refer the reader to one of the many recently published reviews discussed below.

In Delgado [70], the authors develop a taxonomy for characterizing run-time monitoring systems based on their review of publications in the area spanning nearly 25 years. They develop categories based on properties of the specification language used, where and how monitoring is performed, how fault detection events are handled, and their overall applicability to specific applications. They then look at nineteen tools different tools for rum-time monitoring of software faults and classify them according to this taxonomy.

For reviews of the technology itself, Leucker [71] provides an introduction to run-time verification approaches that is highly readable and avoids burdensome mathematical notation. A more formal treatment of run-time verification is provided in Falcone [72], which also illustrates how these can be implemented using several currently available software frameworks. Finally, Bartocci [73] provides an in-depth discussion of run-time verification concepts, addressing the specification of system behavior, construction of monitors, interfaces to extract relevant run-time information, and limitations of the overall approach.

C1.10 RTA Related Concepts

This section discusses a pair of software architectures that are conceptually similar to RTA. These architectures are more general than RTA control architectures in the sense that they were conceived for arbitrary software systems.

C1.10.1 Multi-Version Software

Multi-version techniques use multiple, separately developed software implementations in an attempt to provide fault tolerance. The concept is predicated on the notion that different implementations will tend to exhibit different faults. With the ability to compute outputs of multiple software versions at run-time, it is presumed that at least one of them will produce a correct result, and a selection algorithm can be constructed to identify and return that result Torres-Pomales [74].

Differences between RTA, specifically the Simplex architecture, and multi-version software is addressed in Sha [75], which also contrasts RTA with hardware replication technique. They point out that replication is suitable for providing high reliability against hardware component faults, but that the approach is unsuitable for reliability against software faults, since the same fault will be present in each replicated instance. While multi-version programming may result in software implementations with many non-overlapping faults, they point to research in Knight [76] suggesting that different versions can exhibit a higher frequency of coincident faults than would otherwise be expected. Moreover, these different versions are typically designed to satisfy the same sets of requirements, so that errors in requirements development will be present in all implementations.

In contrast, RTA designs are typically composed of multiple controllers that satisfy different requirement sets, such as high-performance versus high-assurance, as discussed in SectionC1.1. In Sha [77], the author argues that by keeping requirements for a high-assurance controller to the bare minimum necessary to ensure safety, its software implementation can be much simpler than that of a high-performance controller. He analyzes mathematical models of reliability to suggest that, for projects with limited development and assurance budgets, this RTA approach may deliver much higher reliability than a multi-version software approach in which all controllers are designed to satisfy the same requirements.

C1.10.1 Survivability Architectures

A general software architecture for coupling multiple variants of an algorithm that satisfy different sets of requirements is described in Knight [78]. The authors define a *survivability architecture* as one in which one software variant implements a preferred specification that defines full system functionality, and one or more software variants implement alternative specifications that are acceptable under certain adverse conditions. The concept is that if the run-time conditions are such that the preferred specification cannot be met, then an appropriate alternative is adopted that meets the minimum acceptable set of requirements for the current conditions. A full specification of a survivability architecture is defined to include:

1. A set containing the specifications for each software variant, where each specification is a list of requirements the software variant must satisfy;
2. A set of operating environment characteristics (variables) that will affect which software variant is adopted at each point in time;
3. The set of possible value combinations for those environment characteristics;
4. A mapping that indicates the relative value of selecting any one of the specifications given any combination of environment characteristics;
5. An enumeration of the valid transitions between specifications during run-time; and

6. A mapping that indicates the expected reliability of the system under each of the specifications (e.g., the likelihood that each software variant will be unable to meet its specifications).

The authors discuss techniques for using an architecture specification in this form to dynamically switch between software variants at run-time in an attempt to provide the maximum possible value under varying environmental conditions.

C1.11 Loss of Control Prevention

Significant current aerospace research is focused on developing technologies that prevent loss of control (LOC) Wilborn [79] in piloted systems by automatically intervening, either by providing pilot cueing or by directly exerting control over some aspect of the flight. These technologies are similar in many respects to RTA-based designs, in which the manual pilot is viewed as a main controller and the LOC prevention technology constitutes the RTA monitor and backup system. It shares many similarities with the manual-pilot recovery RTA discussed in Section C1.3.

Aircraft LOC has been a longstanding contributor to fatal aviation accidents as noted in Belcastro [80], [81], and [82]. For the period 1999 through 2008, in flight LOC was the largest fatal accident category for commercial transport jets worldwide according to Boeing [83]. An analysis of LOC accidents conducted by NASA researchers, which included “126 LOC accidents (predominantly from Part 121, including large transports and smaller regional carriers) occurring between 1979 and 2009”, found that vehicle upsets contributed to 98 of the 126 accidents per Belcastro [80]. This analysis uses a definition of vehicle upsets from Lambregts [84], which is “any uncommanded or inadvertent event with an abnormal aircraft attitude, rate of change of aircraft attitude, acceleration, airspeed, or flight trajectory.”

Inflight LOC is a major hazard for general aviation (GA) operations as well as Part 121 operations. According to a 2018 Fact Sheet released by the FAA Dorr [85], “Inflight loss of control – mainly stalls – accounts for the largest number of GA fatal accidents.” As Unmanned Air System (UAS) operations expand, evidence is emerging that LOC is a major hazard for these vehicles as well. An analysis of UAS operations published in 2017 found that of the 100 small UAS mishaps investigated, LOC was the largest mishap category, and identified vehicle upset conditions as precursors in 38 of the accidents per Belcastro [86].

Findings on the significance of LOC as a cause of fatal accidents have spurred a variety of research efforts to develop new technologies and training approaches to help reduce LOC. One area of research has been development of onboard systems dedicated to upset recovery, and one class of such systems has strong similarities to RTA architectures. For unmanned systems, issues including reduced situational awareness of the operator and latencies in the command and control links make fully automated upset recovery systems attractive. For vehicles with onboard pilots, both fully automated systems and systems that keep the pilot actively in the loop at a relatively low level are possible, and both approaches have merit. Fully automated systems can react more quickly, but a human pilot may have knowledge and situational awareness that an automated system lacks, leading to better overall outcomes if the human is kept in the loop. The following sections provide a brief overview of systems that have been developed by Barron Associates, one for fully automated upset recovery in unmanned vehicles, and one to aid pilot-in-the-loop upset recoveries in manned vehicles. The systems for manned aircraft provide immediate guidance to pilots when vehicle upsets occur to guide them through a series of recommended control inputs that will return the vehicle to a safe flight condition. With this approach, the pilot retains full control of the aircraft, and may choose to deviate from the recommended actions as necessary, e.g., to avoid a midair collision.

C1.12 Upset Recovery System for UAS

The fully automated upset recovery system developed by Barron Associates [88][89] for UAS is in many ways the most straightforward upset recovery (UR) system variant and has the most obvious parallels to other RTA architectures.

The system is referred to as the RAIDER (Robust Autonomous Integrated Detection and Recovery) system. A high-level block diagram of RAIDER is shown in Figure C-7. The diagram clearly shows that the RAIDER architecture can be considered a variant of the more general RTA architecture. The upset detection block as shown in the figure encompasses both the monitoring and switching components of the generic RTA architecture. The baseline controller depicted in the gray box in Figure C-7 corresponds to the main controller in the RTA architecture. Two backup controllers are included in the system, the Rotational Arrest system, and the Unusual Attitude Recovery system. The Rotational Arrest system is designed, as the name implies, to reduce high angular rates in upset conditions include spins, but also to reduce large aerodynamic angles. After angular rates have been reduced adequately, the vehicle may well remain in an upset condition, and control is transferred to the Unusual Attitude Recovery backup controller to maneuver the vehicle to a condition that is close to straight and level and at an appropriate airspeed for the flight condition. This stage consists of a robust inner-loop control law that is configured to guide the vehicle back to straight-and-level flight very reliably. At this point it is expected that the baseline controller will be able to safely resume control of the vehicle, which implies that the baseline controller should include capabilities to handle off-nominal vehicle dynamics.

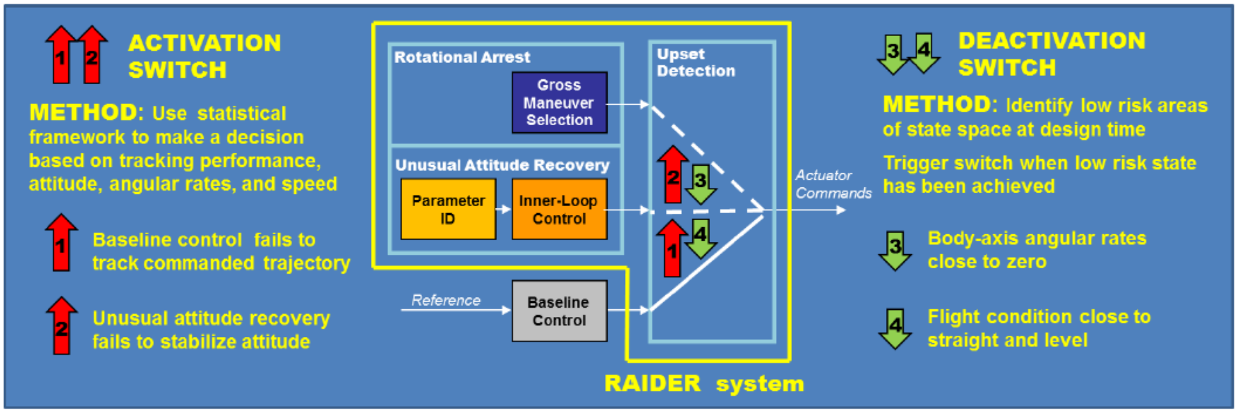

1
2
ACTIVATION SWITCH
METHOD: Use statistical framework to make a decision based on tracking performance, attitude, angular rates, and speed
1
Baseline control fails to track commanded trajectory
2
Unusual attitude recovery fails to stabilize attitude
Rotational Arrest
Gross Maneuver Selection
Unusual Attitude Recovery
Parameter ID
Inner-Loop Control
Reference
Baseline Control
Upset Detection
2
3
1
4
Actuator Commands
RAIDER system
3
4
DEACTIVATION SWITCH
METHOD: Identify low risk areas of state space at design time
Trigger switch when low risk state has been achieved
3
Body-axis angular rates close to zero
4
Flight condition close to straight and level

Figure C-7. Upset Detection System for Activation/Deactivation of RAIDER Stages

The decision about when to activate each stage of the recovery is difficult to make at design-time. UAS, especially small UAS, often lack sufficient aerodynamic data to make offline analysis of coverage provided by each stage feasible. In addition, upsets are often the result of off-nominal precipitating factors that are hard to comprehensively represent at design-time.

The upset detection system interfaces with the two-stage recovery architecture and determines at run-time when to switch into (activation switch) and switch out of (deactivation switch) each stage of the recovery process. Inspecting the left side of Figure C-7, the activation switches are represented by red arrows and include:

1. Switch #1: the switch from the baseline control stage to the unusual attitude recovery stage; and
2. Switch #2: the switch from the unusual attitude recovery stage to the rotational arrest stage.

To trigger each activation switch, the upset detection system uses a statistical decision-making framework to decide *whether an upset occurred*, and if so, *which recovery stage should be used to affect the most rapid recovery*.

The upset detection system leverages a framework based on statistical decision theory. A Bayesian estimator combines information from sensor data and control inputs with information embedded in a model of expected dynamics to estimate the output of various vehicle states. Maximum likelihood tests conducted using probabilities generated by the Bayesian estimator identify discrepancies between observed vehicle states (based on sensor data) and expected vehicle states (based on model predictions). The tests are used to accept or reject the null hypothesis: The closed-loop response of the vehicle is consistent with the expected closed-loop response. The result of the maximum likelihood tests is synthesized with other accepted measures of upset and loss of control to make a final determination on when to activate the RAIDER system.

The statistical framework allows the decision-making process to effectively combine numerous pieces of information including closed-loop tracking performance, pitch and roll attitude, angular rates, and airspeed. The decision-making framework emphasizes the closed-loop response of higher-level vehicle states (such as Euler angles, airspeed, and altitude) to determine when to activate the unusual attitude recovery stage and lower-level vehicle states (such as body-axis angular rates and aerodynamic angles) to determine when to activate the rotational arrest stage. While not explicitly shown in the figure, depending on the severity of the upset, it may be necessary to trigger a switch from baseline control directly to the rotational arrest stage.

As the recovery progresses, the upset detection system uses different criteria to trigger the deactivation switches as shown on the right side of Figure C-7. The deactivation switches are represented by green arrows and include:

1. Switch #3: the switch from the rotational arrest stage to the unusual attitude recovery stage (which typically occurs when body-axis angular rates are close to zero); and
2. Switch #4: the switch from the unusual attitude recovery stage to the baseline control stage (which typically occurs when the vehicle is close to straight-and-level flight).

The exact decision criteria (i.e., the definition of “close”) for deactivating the upset recovery system are established at design-time based on an established envelope for normal operations of the specific vehicle on which the system is being deployed.

C1.12.1 Upset Recovery with an Onboard Pilot

The approach described above for UAS could be applied to vehicles with an onboard pilot. Advantages of this strategy include the potential for very fast response time, precise execution of the recommended recovery strategies, and the flexibility to use command sequences (e.g., those with significant high frequency content) that would be difficult for a pilot to input. This section describes an alternate approach in which the onboard pilot retains full control of the vehicle, and the upset recovery system computes and displays to the pilot (through visual, haptic, and/or other means) a recommended recovery strategy. This approach is advantageous in that it enhances the situational awareness of the pilot, promotes rapid and appropriate responses to upset conditions, and keeps the pilot in the loop, e.g., to appropriately prioritize between collision avoidance and upset recovery. In this approach, the pilot can be thought of as a main controller, the behavior of which is modified (rather than being replaced) through the addition of cues provided by the upset recovery system when it is active. This differs slightly from many RTA designs in which a main controller is replaced by some backup control approach, but the upset recovery approach retains the typical RTA monitoring and switching component in the form of an upset detection system and provides backup control configurations that are activated when problems are detected.

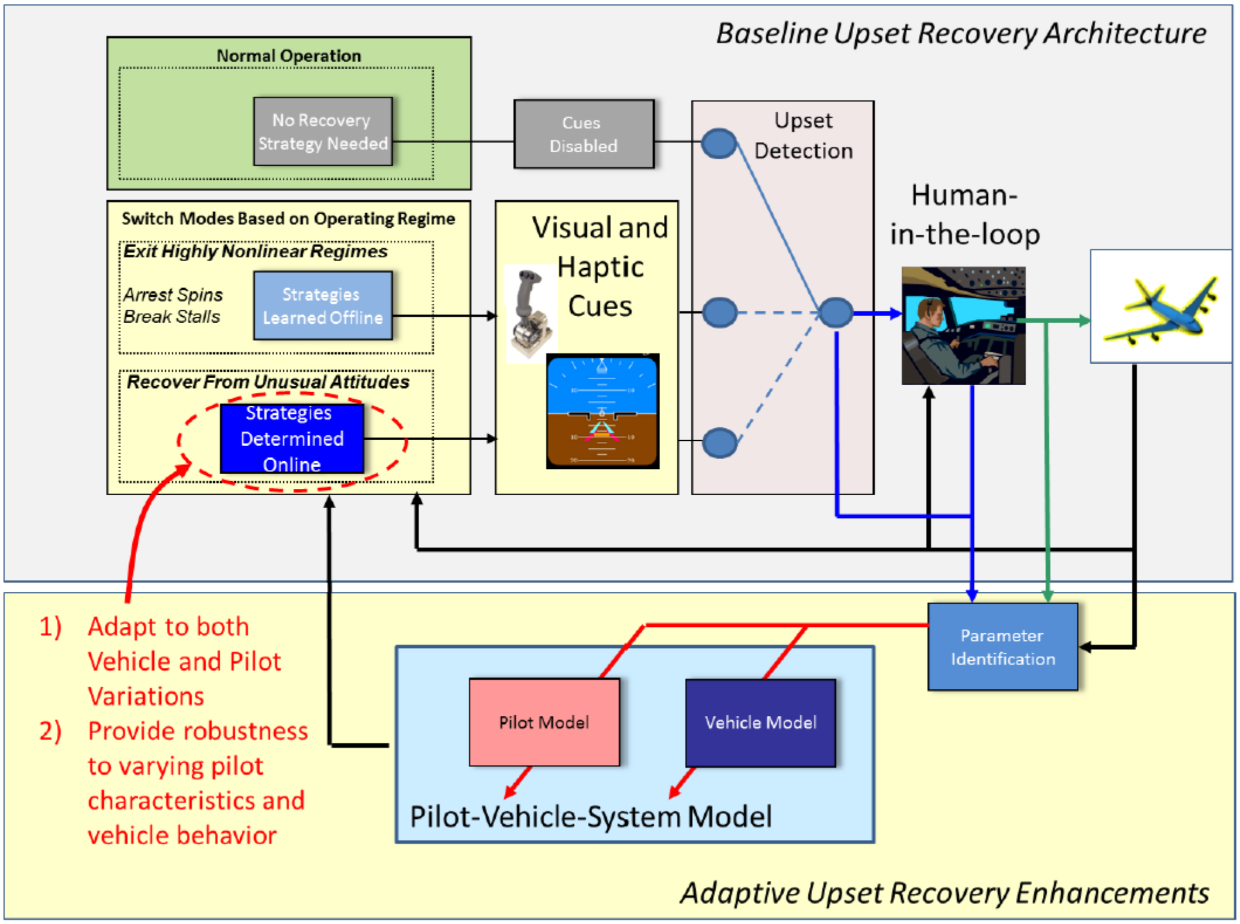

Baseline Upset Recovery Architecture
Normal Operation
No Recovery Strategy Needed
Cues Disabled
Upset Detection
Human-in-the-loop
Switch Modes Based on Operating Regime
Exit Highly Nonlinear Regimes
Arrest Spins
Break Stalls
Strategies Learned Offline
Visual and Haptic Cues
Recover From Unusual Attitudes
Strategies Determined Online
1) Adapt to both Vehicle and Pilot Variations
2) Provide robustness to varying pilot characteristics and vehicle behavior
Parameter Identification
Pilot Model
Vehicle Model
Pilot-Vehicle-System Model
Adaptive Upset Recovery Enhancements

Figure C-8. Upset Recovery Design for Vehicles with Onboard Pilot

Figure C-8 shows two variants of the upset recovery system developed by Barron Associates for vehicles with an onboard pilot. The basic system [Gandhi [87]] does not explicitly adapt to off-nominal conditions, though recovery strategies are designed with a goal of robustness to accommodate both aerodynamic uncertainties, which are likely to grow when the vehicle departs the normal flight envelope, and off-nominal vehicle responses. The enhancement depicted in the lower portion of Figure C-8 adds explicit adaptation to off-nominal pilot-vehicle-system behavior [Richards [88] & [89]. The complete system including the adaptive enhancement is referred to as DAGUR (Damage Adaptive Guidance for Upset Recovery).

C1.12.2 Upset Detection

In the RAIDER system, failure of the inner-loop controller to accurately track commands is a key indicator of an upset condition. When an onboard pilot is flying, it is not possible to know the pilot's intent at any specific time, and instead the upset detection component must look at more general indicators of vehicle upset and loss of control. This approach has limitations, for example, a very aggressive maneuver to avoid a collision might be difficult or impossible to discriminate from a vehicle upset. Because the pilot remains in the loop, however, the pilot may choose to ignore upset recovery guidance when conditions warrant, and the guidance may be use of in returning to a safe flight condition following an aggressive maneuver even if that maneuver was intentional.

Barron Associates' upset detection system implementation for manned vehicles draws on Wilborn [79][90], in which five quantitative loss-of-control (QLC) envelopes were identified. These envelopes were defined in terms of key aircraft states and are summarized in the following list:

- **Adverse Aerodynamics (AA) Envelope** – Function of normalized angle-of-attack (α_{norm}) vs. normalized sideslip (β_{norm}).
- **Unusual Attitude (UA) Envelope** – Function of pitch angle (θ) vs. roll angle (ϕ).
- **Structural Integrity (SI) Envelope** – Function of load factor (n) vs. normalized airspeed (V_{norm}).
- **Dynamic Pitch Control (DPC) Envelope** – Function of dynamic pitch angle (sum of the current pitch angle and its derivative) vs. percent pitch control.
- **Dynamic Roll Control (DRC) Envelope** – Function of dynamic roll angle (sum of the current roll angle and its derivative) vs. percent roll control.

According to Wilborn [79], excursions outside three or more envelopes are a reliable indication of loss of control. The authors note that "the interval between the time of the first envelope excursion and the time when control was lost, defined as the critical window, quantifies how long the crew had between the initial upset and the resulting loss of control to correct the situation." The upset recovery system can provide guidance in that critical window, activating, e.g., with the first excursion of one or two envelopes. These envelopes form the basis of the upset detection system, though the specific envelope parameters and the exact set of activation criteria may vary based on the vehicle.

Vehicle damage and failures are also an important consideration in the design of the upset detection system. In these cases, safety margins are typically reduced, and it makes sense to enforce a stringent criterion for upset detection. For instance, one envelope excursion might trigger the system in a failure condition rather than two envelope excursions for a healthy vehicle.

C1.12.3 Closed-Loop Recovery Module

Figure C-9 depicts the pilot vehicle system with the upset recovery guidance system active, and including the adaptive enhancement to that system. The recovery guidance module accepts a desired reference command (x_r), corresponding to a safe flight condition, state feedback from the vehicle (x), and, optionally, vehicle parameter estimates. The recovery guidance module continuously outputs recommended stick (or wheel/column), pedal, and throttle commands (δ_R) that will to lead the Pilot-Vehicle-System (PVS) to a safe flight condition. By continuously generating these commands based on the current aircraft state (closed-loop recommendations), the system ensures that the pilot can start and stop following the provided guidance at any time without compromising the validity of the current guidance.

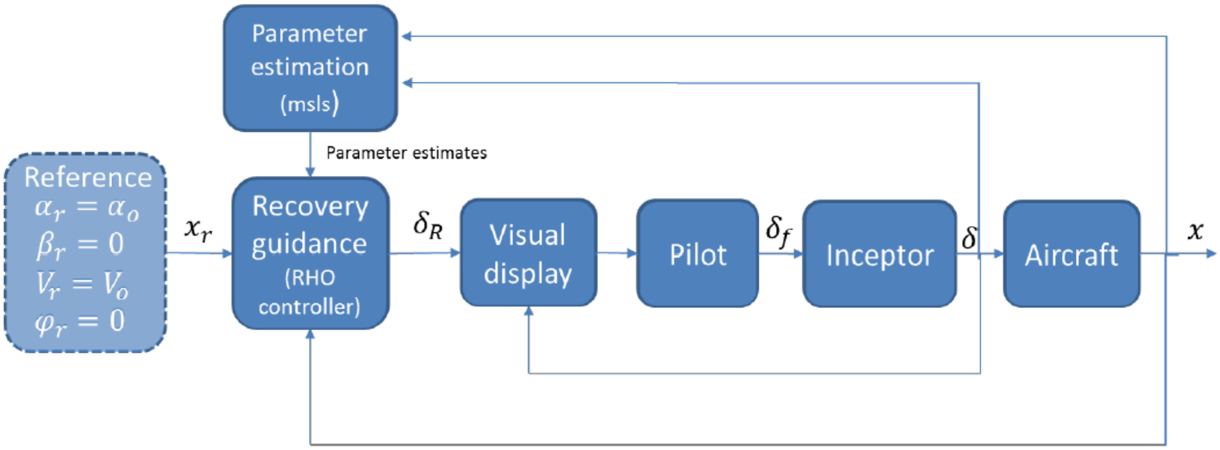


Figure C-9. Pilot Vehicle System with Active Upset Recovery and Adaptation

A critical consideration in design of the recommended recovery strategy is that this strategy is an input not to the vehicle alone, but to the combined pilot vehicle system. The behavior of the pilot must be explicitly considered in the design of the recovery strategy in order to reliably achieve the desired outcomes.

An important aspect of the research effort to develop DAGUR was work to map psychological and physiological conditions such as distraction and high stress into the existing pilot modeling theory. Also, the team built upon prior pilot behavior findings for open-loop recoveries Richards [90] to examine the impact of these pilot dynamics on the closed-loop recoveries considered here.

C1.13 Certification of RTA-Based Systems

The possibility of RTA based architectures appearing in small aircraft is contemplated by the FAA's Small Airplane Standards Branch. From SAIL [91]:

> *If the applicant proposes to use Run - Time Assurance/Health Monitoring Executive Systems, then they must coordinate their proposals with the Small Airplane Standards Branch to determine the level of FAA involvement with respect to policy and guidance. Run - Time Assurance/Health Monitoring Executive Systems are high - level monitoring and protection systems and are new to the Part 23 fleet. The FAA is actively working on draft policy and guidance for the use of this new and novel technology.*

A small number of publications exist that specifically address the certification challenges associated with RTA-based designs. Many begin with the assumption that in the near future there will be a significant push for vehicles that include advanced control systems with features such as adaptation, reconfiguration, intelligent autonomy, and real-time machine learning. These features are typically intended to provide higher levels of performance and/or robustness than classical controllers. However, their behavior at any point in time is a complex function of the operational and environmental conditions which they have previously experienced. Designers cannot know every possible state or outcome such systems will exhibit when exposed to the infinite possibilities of real-world scenarios, unforeseen events, and unanticipated conditions. As a result, it can be difficult to define performance and safety requirements for these systems, and they are often impossible to fully analyze at design time to ensure that they will meet those requirements.

In Jacklin [92], the authors point out that under current regulations, an ability to completely specify requirements is assumed. Systems must be shown to meet their intended function, and compliance with this fundamental tenet implies that all necessary and relevant requirements must be explicitly derived and that for each valid input there must be a defined intended response. They go on to look at a number of specific validation and verification challenges associated with adaptive systems, including limitations of formal methods techniques, limitation in techniques for proving stability, limitations in fault detection capabilities, and difficulties in generating sufficient test cases.

In more recent publications, Wilkinson 2013 [93] and [94] look at software assurance requirements for adaptive systems from the perspective of DO-178B and DO-178C. They describe a variety of adaptive algorithms, their applicability to aircraft control, and likely safety issues associated with their use. The authors point out that a primary difficulty in certifying adaptive systems is that they change software parameters at run-time in response to operating conditions. However current assurance processes implicitly assume that such parameters are statically determined at design-time. While gain-scheduled designs are commonly used to cover an entire flight envelope, these typically have a small number of parameter-value combinations, which can be explicitly enumerated. In contrast, adaptive systems typically have one or more continuous-valued parameters that can take on an essentially infinite number of value combinations. As a result, it is difficult to show that the system will satisfy all of its requirements under all possible parameter values. Moreover, it is difficult to define up front exactly what the system behavior should be in any particular test because the state of the adaptation component will be unknown, in general.

In light of this, they consider what would be required to certify an aircraft with an adaptive controller. While they don’t explicitly refer to the design as run-time assurance, they do explore a system design that pairs an adaptive controller with a traditional gain-scheduled controller. They offer a number of recommendations, for both requirement validation and verification. For requirement derivation and validation, they recommend that requirements analysis should be preceded by the construction of an evidence-based safety case, from which safety objectives would be derived. In turn, these safety objectives would be used to derive system properties addressing when an adaptive component may be engaged and disengaged, as well as the bounds that should be placed on learned parameter values. They further recommend that the standard ARP4761 [23] be updated to included evidence-based safety case methodologies. For verifying the requirements of adaptive systems, they recommend the use of model-based development techniques using mathematical models to express desired safety properties. Formal methods would then be used to verify those properties, along with existing test and simulation techniques.

In Bhattacharyya [95], the authors discuss upset recovery, catastrophic damage, and autonomous operations as example applications in which adaptive control techniques are potentially valuable. They survey a number of adaptive control and related artificial intelligence algorithms, and they discuss the certification challenges associated with their use. The authors reach conclusions that are very similar to papers cited above regarding the difficulty of deriving comprehensive and verifiable requirements for adaptive systems. Additionally, they point out that adaptive algorithms frequently embed dynamical models, numerical solvers, and other algorithms that significantly increase their complexity. It can be exceedingly difficult for reviewers to fully grasp the resulting design, and it can be very difficult to demonstrate the absence of unintended functionality.

The report contains a number of strategies that could mitigate these challenges, and categorize specific adaptive algorithms based on the mitigations that would enable their use. They argue that some adaptive control algorithms are benign enough that educational efforts to bridge gaps in understanding and vocabulary between developers and regulators should be sufficient to enable certification, pointing to L1 adaptive control as an example. They make the point that inconsistent use of terms between these communities can lead to the perception of certification barriers for these algorithms, even if those algorithms could be certified under existing standards. For other, more complex adaptive control algorithms, the authors suggest that with some minor modifications of existing certification standards, adaptive controllers used in conjunction with run-time assurance architectures could be certified. The standards would need to allow functions with lower levels of demonstrated assurance to be used in a system designed to meet high assurance objectives. This would entail an ability to reason about time-varying assurance levels.

In Goodloe [96], the author catalogs several process and design objectives for RTA-based designs that would support their certification. While not explicitly addressing the regulatory framework, these objectives could be very useful as high-level goals in an assurance case that explicates the certification argument. This analysis focuses primarily on the RTA monitoring component. Restated in the form of assurance case goals, these objectives are:

1. Requirements for the RTA monitor are derived from validated system-level requirements.
2. Required behavioral properties of the RTA monitor are expressed in a formal logic.
3. All state and environment variables needed by the RTA monitor are observable.

4. There is bi-directional traceability between RTA monitor requirements and implementation code.
5. The main controller and RTA monitor are not subject to common mode failures.
6. The RTA monitor's activities do not compromise the safety of either the main or backup controllers.
7. The RTA monitor specification is correct.
8. The RTA monitor software correctly implements its specification.

The author provides examples of tool-level support for each of these objectives that is provided by the Copilot framework (Pike [97]).

The recently published ASTM F3269-17 [98] provides guidance on certification of RTA-based designs for unmanned aircraft, including beyond visual line of sight. While its focus is on unmanned aircraft, it acknowledges the potential application to other aviation operations. This standard specifically addresses only RTA systems in which the main controller has advanced, difficult to certify capabilities and the backup controller follows a traditional design (referred to as main-adaptive in the discussion above).

ASTM F3269-17 [98] defines a consistent set of terms for referring to RTA functions, subcomponents, and timing. It then presents a hierarchy of requirements for the RTA architecture as a whole and for each of its subcomponents. These requirements begin with a prioritization and partitioning of functions within the RTA architecture, followed by an operational risk assessment that is subsequently used to assign a safety criticality to each subcomponent and to derive required RTA timing characteristics. It separately addresses requirements for RTA system inputs and their management, the backup controller, and the RTA monitor and switch. The standard places only minimal requirements on the main controller, stating that it should be partitioned from the other RTA components and that it should receive inputs and generate output commands that are sufficient to ensure intended function and adequate performance of the vehicle being controlled. The standard also addresses documentation that should be provided to support certification. This includes quantitative probabilities of failure for critical components, as determined by FMEA.

A history of this standard and the philosophy adopted by the ASTM F38 committee that developed it is contained in Cook [99].

C1.14 Supporting Techniques

Within the publications cited in Section C1.13, several techniques are mentioned that may be useful for supporting certification efforts for systems that employ RTA architectures. In this section, we provide brief introductions to these techniques.

C1.14.1 Formal Methods

Formal methods are a class of techniques that attempt to rigorously prove whether a dynamical system will or will not exhibit behaviors of interest. They require that the system be described by a mathematical model and that the behaviors of interest are formally stated in an unambiguous logical calculus. Automated tools are typically employed to prove or disprove those formal statements.

The literature on formal methods is extensive and stretches back more than 40 years. Because of the breadth of available techniques, they will not be surveyed in this report. Rather, we point the reader to an excellent survey contained in Clark [100], which discusses many formal methods technologies specifically in the context of RTA-based designs. This survey covers nearly 300 relevant publications, providing an overview of key topics, including:

1. Formal methods for verifying whether a hybrid system satisfies a desired specification, and how those methods can be leveraged in the design of RTA systems;
2. Applications of run-time verification techniques to boundary computation and switching logic for RTA monitors;
3. Timing considerations in RTA designs; and
4. Model-based design and testing for RTA architectures.

C1.14.2 Uncertainty Quantification and Probabilistic Analysis

Uncertainty quantification (UQ) is an area of engineering analysis that seeks to quantify the effects of parametric uncertainty on results predicted from mathematical models. A widely adopted UQ technique is to develop a surrogate model that defines a functional relationship between the uncertain parameters and the predicted response. Many recent publications have leveraged generalized polynomial chaos (gPC) theory for constructing these surrogate models. This theory is particularly powerful in the context of UQ because:

- It is supported by a rigorous and well-developed theoretical foundation.
- It has a very compact representation that can characterize dependence on scalar-valued, vector-valued, and function-valued quantities.
- It can represent essentially arbitrary weighting functions, including continuous probability distributions for addressing probabilistic uncertainty.
- It naturally handles arbitrary dependency structures between all quantities in a model.
- It has an adjustable-complexity, allowing an analyst to make tradeoffs between the accuracy of a result and the computational resources expended to compute it.

In many ways, gPC-based techniques can be thought of as complementary to formal methods techniques. Whereas formal methods attempt to prove that system-defined quantities, will or will not take certain values given a range of uncertain inputs, gPC techniques attempt to characterize how those uncertainties propagate through the system to affect other system quantities. gPC theory is frequently applied in a probabilistic context, in which uncertain parameters are assumed to be governed by probability distributions. However, an assumption that uncertainties are inherently probabilistic is not required. When used for deterministic analyses, these probability distributions can be defined as uniform over all possible uncertainty combinations. Without loss of generality, the discussion that follows will assume probabilistic uncertainties.

gPC theory was originally developed in Wiener [101] for the study of homogeneous turbulence. The completeness properties and computational tractability of gPC have lead many engineers and scientists to apply it in a wide range of domains. At its most fundamental level, gPC theory is concerned with universal representations of random variables. By universal, we mean that any finite-variance random quantity can be approximated to arbitrary accuracy. The gPC representation hinges on the fact that, in probability theory, random processes are defined as functions over a set of possible outcomes, and that well-behaved functions can be expressed as a generalized Fourier series. In the gPC representation, the set of possible outcomes is abstractly characterized by the possible values of a simple random vector, and an arbitrary random process, at each point in time, is a polynomial function of this random vector.

It is well known that for any probability density function $f(z)$, there is a unique family of polynomials $\{\psi_k\}$ such that:

1. Each $\psi_k(\cdot)$ is a polynomial of order k; and
2. The family is orthonormal with respect to the weighting function $f(z)$.

The orthonormality criterion means that:

$$E\big[\psi_i(Z)\psi_j(Z)\big] = \int_{-\infty}^{\infty} \psi_i(z)\psi_j(z)f(z)\,dz = \begin{cases} 1 & if\ i = j \\ 0 & otherwise \end{cases} \quad \text{Eq 23}$$

Note that from these two properties, it follows that $\psi_0(Z) = 1$.

The above polynomial family can be extended to vectors $\boldsymbol{Z} = [Z_1, Z_2, \dots, Z_D]^T$ with probability density function:

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \prod_{d=1}^{D} f_{z_d}(z_d) \quad \text{Eq 24}$$

In this case, we define the polynomial index $\boldsymbol{k} = [k_1, k_2, \dots, k_D]^T$ as a vector indicating a corresponding product of univariate polynomials:

$$\psi_{\boldsymbol{k}}(\boldsymbol{z}) = \prod_{d=1}^{D} \psi_{d,k_d}(z_d) \quad \text{Eq 25}$$

where the polynomials $\{\psi_{d,i}\}$ are an orthonormal family with respect to the weighting function $f_{\boldsymbol{Z_d}}(\cdot)$. It is straightforward to show that the family of multivariate polynomials $\{\psi_{\boldsymbol{k}}\}$ defined in this way is orthonormal with respect to the joint density $f_{\boldsymbol{Z}}(\boldsymbol{z})$.

The gPC technique uses polynomial families such as these to represent random quantities. More specifically, according to gPC theory, any random variable G with finite variance can be expressed as a polynomial function of a random vector $\boldsymbol{Z}$ as:

$$G = l.i.m._{K\to\infty} \sum_{\{\boldsymbol{k}:0\le k_d \le K\}} g_k\, \psi_{\boldsymbol{k}}(\boldsymbol{Z}) \quad \text{Eq 26}$$

Where $l.i.m.$ stands for *limit in the mean*. The coefficients $g_{\boldsymbol{k}}$ are constant values found by projecting G on to the corresponding basis functions, as:

$$g_{\boldsymbol{k}} = E[G\psi_{\boldsymbol{k}}(\boldsymbol{Z})] = \int_{-\infty}^{\infty} G\psi_{\boldsymbol{k}}(\boldsymbol{z}) f_{\boldsymbol{Z}}(\boldsymbol{z})\,d\boldsymbol{z} \quad \text{Eq 27}$$

In the UQ applications, components of the vector $\boldsymbol{Z}$ are the uncertain parameters that affect a system. Quantities like G represent response variables of the system, such as states, outputs, performance metrics, safety metrics, etc. These response variables are explicitly modeled as polynomial functions of the uncertain parameters. The quantities $g_{\boldsymbol{k}}$ are (generalized) coefficients in those polynomial representations. By computing the $g_{\boldsymbol{k}}$, you get a complete characterization of the functional relationship between uncertainties and responses.

In practice, the infinite series representing G will be truncated to a finite number of terms. This truncation level provides a means of adjusting the accuracy of the representation and, consequently, the amount of computational effort that must go into computing results.

C1.14.3 Assurance Case Approaches

In its most basic form, an assurance case is a structured argument asserting that a system exhibits specific properties regarding its safety, performance, security, or other aspects. Structure is provided by decomposing such an argument into a hierarchical arrangement of claims, each of which is supported by evidence. The top level of the claim hierarchy represents the property assertion being addressed in the argument. Claims at subsequent levels support the claims above, in a deductive reasoning chain, down to individual and clearly substantiated terminal assertion.

The Goal Structuring Notation (GSN), a standard for graphically representing assurance cases, is defined in ACWG [102]. It defines the core notation, describes available extensions, and provides guidance on the development of assurance cases.

Numerous publications discuss assurance case techniques and their applications. Rather than review this extensive literature, we refer the reader to the extensive discussion of assurance case practice presented in Reinhart [103]. The authors describe common notations, explore techniques for evaluating assurance cases, and provide numerous examples.

The publication by Rushby [104] is particularly relevant to run-time assurance architectures. In it, he author points out that assurance cases and run-time verification approaches can be complementary to each other. That is, assurance cases are decomposed into a set of claims, which are natural choices for run-time verification of any properties that cannot be rigorously proven at design-time. Correspondingly, the fact that those properties are explicitly checked provides evidence to support the corresponding claim in the assurance case.

Appendix D: Government Furnished Information

This appendix contains a summary of Government Furnished Information (GFI) regarding assurance practices and alternate assurance concepts. References and reports were provided by Wilfredo Torres-Pomales and Kurt Woodham, NASA Langley Research Center.

Title	Functional failure path analysis of airborne electronic hardware
Authors	S.C. Beland Boeing Co., Seattle, WA, USA
Date	October 2000
Availability	19th DASC. 19th Digital Avionics Systems Conference. Proceedings (Cat. No.00CH37126)
Abstract	RTCA DO-254, "Design Assurance Guidance for Airborne Electronic Hardware," outlines an approach for providing design assurance confidence for functions of a range of design assurance levels. In order to decompose the hardware functions, DO-254 provides a brief overview of Functional Failure Path Analysis (FFPA) but is limited in detail and provides no examples. This paper presents a detailed explanation of FFPA and illustrates it with a representative example.

Title	Considerations in Assuring Safety of Increasingly Autonomous Systems
Authors	Erin E. Alves, Devesh Bhatt, Brendan Hall, Kevin Driscoll and Anitha Murugesan Honeywell International, Inc., Golden Valley, Minnesota John Rushby SRI International, Menlo Park, California
Date	July 2018
Availability	https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20180006312.pdf
Abstract	This report is the product of work performed on the Assurance Reasoning for Increasingly Autonomous Systems (ARIAS) project, funded by NASA under Contract: NNL16AA06B and Task Order: NNL16AA96T. The ARIAS project objective is to perform the following tasks in the context of reduced crew operations: 1. Identify safety requirements introduced when the human role in systems operation is reduced, changed or eliminated, by an IA system. 2. Identify changes necessary in systems safety assurance to account for changes to human roles (e.g. task responsibilities and decision-making authority). 3. Identify changes necessary in system architectures to provide an equivalent, or better, level of safety for IA systems. 4. Identify new Verification and Validation (V&V) capabilities needed to assure IA systems.

Title	EASA Special Condition for VTOL
Authors	EASA
Date	July 02, 2019
Availability	https://www.easa.europa.eu/document-library/product-certification-consultations/special-condition-vtol

Abstract	The Agency has received a number of requests for the type certification of vertical take-off and landing (VTOL) aircraft, which differ from conventional rotorcraft or fixed-wing aircraft. In the absence of certification specifications for the type certification of this type of product, a complete set of dedicated technical specifications in the form of a special condition for VTOL aircraft has been developed. This special condition addresses the unique characteristics of these products and prescribes airworthiness standards for the issuance of the type certificate, and changes to this type certificate, for a person-carrying VTOL aircraft in the small category, with lift/thrust units used to generate powered lift and control.

Title	Hazards Analysis and Failure Modes and Effects Criticality Analysis (FMECA) of Four Concept Vehicle Propulsion Systems
Authors	Patrick R. Darmstadt, Ralph Catanese, Allan Beiderman, Fernando Dones, Ephraim Chen, Mihir P. Mistry, Brian Babie, Mary Beckman, and Robin Preator The Boeing Company, Philadelphia, Pennsylvania
Date	June 2019
Availability	https://ntrs.nasa.gov/search.jsp?R=20190026443
Abstract	The primary objective of this research effort is to identify failure modes and hazards associated with the concept vehicles and to perform functional hazard analyses (FHA) and failure modes and effects criticality analyses (FMECA) for each. Boeing also created a Fault Tree Analysis (FTA) for each of the concept vehicles, as the FTA contains the connectivity between systems and is an accepted, top-down method to analyze the safety of an air-vehicle. Conceptual design of notional powertrain configuration for each of four (4) NASA RVLT Concept Vehicles were developed in as much detail as was necessary to support the reliability and safety analysis for this project. Functional block diagrams from each of the conceptual powertrain configurations were created and used to order the FHA, FMECA, and FTA. Hazards were identified and the severity of each were categorized in the FHA for use in a follow-up FMECA. The FTA took inputs from the FMECA and the functional block diagrams to develop the connectivity and develop a quantitative architecture that could be used to perform sensitivity studies, as related to vehicle safety. Guidelines for reliability targets for both the air vehicle and the operation in the UAM mission are discussed. An industry literature search was performed in order to assess gaps in existing government regulations and industry specifications. The industry literature search led to air-vehicle and operational reliability discussions, as related to Distributed Electric/Hybrid-Electric Propulsion (DE/HEP) system operating in the UAM role. A discussion of results and recommendations for future work is also provided.

Title	FAA EZ Fly Concept May Lead to Simpler Piloting for Urban Mobility
Authors	Frank Wolfe
Date	December 4, 2018
Availability	https://www.rotorandwing.com/2018/12/04/faa-ez-fly-concept-may-lead-simpler-piloting-urban-mobility/
Abstract	The FAA is undertaking an EZ Fly Aircraft and Demonstrator project to simplify piloting of small aircraft and reduce general aviation accidents, and the concept may eventually advance the remote piloting of fleets of urban mobility aircraft. EZ Fly will demonstrate a simplified flight path-based Advanced Flight Control System, while using a fusion of sensors, control laws, displays, and a simplified pilot interface with full-envelope protection. The FAA is conducting the EZ Fly research to develop a means of compliance to certify similar systems. FAA representatives discussed the EZ Fly concept last week with aerospace industry officials at a meeting of the Simplified Vehicle Operations (SVO) panel of the General Aviation Manufacturers Association (GAMA) at Embry-Riddle Aeronautical University in Daytona Beach, Florida.

Title	Technology Independent Assurance Method
Authors	Mike DeWalt, Federal Aviation Administration, Reston, Washington G. Frank McCormick, Certification Services, Inc., Eastsound, Washington
Date	33rd Digital Avionics Systems Conference October 5-9, 2014
Availability	https://ieeexplore.ieee.org/document/6979529
Abstract	Avionics software and hardware must, like all engineered products, be fit for use. The recently published SAE ARP4754A [1], “Guidelines for Development of Civil Aircraft and Systems,” addresses this obligation in the context of system development. Model-Based Development has been widely adopted in civil aeronautics and blurs traditional distinctions between systems-engineering disciplines on the one hand and software/hardware disciplines on the other. One consequence of that blurring has been loss of clarity in the guidance associated with design assurance. This paper seeks to resolve such questions. The Technology Independent Assurance Method (TIAM) is proposed as a straightforward extension of previous engineering strategies and is intended to span a given product's development cycle from concept to production readiness. TIAM generalizes the recognition that development is a progressive activity with an initially unknowable number of successive refinements that proceed from the more abstract to the less abstract. TIAM identifies, articulates, and preserves the integrity of each intellectual abstraction along that path, highlighting all required relationships of each abstraction to its predecessors and successors. At each step, developers are free to choose appropriate processes and tools. This paper provides an overview of TIAM. In addition, this paper illustrates TIAM's applicability to certain real-world challenges arising from Model- Based

	Development as currently applied to airborne hardware and software by systems engineers

Title	Disengagements: Wrong Metric for AV Testing
Authors	Junko Yoshida
Date	4/10/2019
Availability	https://www.eetimes.com/disengagements-wrong-metric-for-av-testing/#
Abstract	It's been more than a year since the first fatal accident caused by a self-driving testing vehicle in Arizona cast a shadow over this heavily hyped technology. In the fast-moving tech world, the death of Elaine Herzberg, struck down by an Uber autonomous test vehicle, seems like old news, faded from most people’s memory. But for safety expert Phil Koopman, associate professor at Carnegie Mellon University and co-founder of Edge Case Research, this tragedy has triggered new research, leading him to question whether the companies testing these vehicles are designing an effective safety test platform.

Title	Autonomous Vehicle Safety Technical and Social Issues
Authors	Prof. Philip Koopman, Carnegie Mellon University
Date	Sept. 18, 2018
Availability	https://users.ece.cmu.edu/~koopman/pubs/koopman18_waise_keynote_slides.pdf
Abstract	A Doer-Checker safety-envelope-based architecture pattern similar to RTA is being studied for autonomous vehicle safety. See slides 8 and 9.

Title	Explicate ’78: Assurance Case Applicability to Digital Systems DOT/FAA/TC-17/67
Authors	C. Michael Holloway & Patrick J. Graydon
Date	January 2018
Availability	http://www.tc.faa.gov/its/worldpac/techrpt/tc17-67.pdf
Abstract	This report documents the results of the Explicate '78 project. The project was conducted by NASA Langley Research Center in support of an annex (Assurance Case Applicability to Digital Systems) to the Reimbursable Interagency Agreement IA1-1073 (Design, Verification, and Validation of Advanced Digital Airborne Systems Technology). In particular, the report describes an assurance case developed to express the arguments contained in, or implied by, DO-178C (Software Considerations in Airborne Systems and Equipment Certification), which implicitly justifies the assumption that the document meets its stated purpose of providing “guidelines for the production of software for airborne systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements.” An appendix to the

	report provides an assurance case for DO-330 (Software Tool Qualification Considerations).

Title	Explicate '78: Uncovering the Implicit Assurance Case in DO–178C
Authors	C. Michael Holloway, NASA Langley Research Center
Date	February 3, 2015
Availability	https://ntrs.nasa.gov/search.jsp?R=20150009473
Abstract	Abstract For about two decades, compliance with Software Considerations in Airborne Systems and Equipment Certification (DO–178B/ED–12B) has been the primary means for receiving regulatory approval for using software on commercial airplanes. A new edition of the standard, DO–178C/ED–12C, was published in December 2011, and recognized by regulatory bodies in 2013. The purpose remains unchanged: to provide guidance 'for the production of software for airborne systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements.' The text of the guidance does not directly explain how its collection of objectives contributes to achieving this purpose; thus, the assurance case for the document is implicit. This paper presents an explicit assurance case developed as part of research jointly sponsored by the Federal Aviation Administration and the National Aeronautics and Space Administration.

Title	Certification Authorities Software Team (CAST) Position Paper CAST-5 Guidelines for Proposing Alternate Means of Compliance to DO-178B
Authors	Certification Authorities Software Team
Date	June 2000
Availability	https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast_papers/media/cast-5.pdf
Abstract	The purpose of this paper is to provide guidelines for industry in proposing alternate means and for the certification authorities and designees to evaluate the feasibility of those proposed alternate means for meeting the safety objectives of the regulations.

Title	CityAirbus Will Build An Understanding Of Technologies Needed For UAM
Authors	
Date	February 28, 2019
Availability	Aviation Week and Space Technology
Abstract	

Title	CityAirbus Prototype Unmanned Air Taxi to Take Flight in March

Authors	Dan Parsons
Date	February 28, 2019
Availability	https://www.rotorandwing.com/2019/02/28/cityairbus-prototype-unmanned-air-taxi-take-flight-march/
Abstract	DONAUWORTH, Germany — CityAirbus, the namesake company's prototype unmanned air taxi, should break contact with the ground sometime in March and then embark on a flight test campaign to incrementally expand the two-ton vehicle's envelope and learn its capabilities.

Title	Development of Powered-Lift Airworthiness Standards as Applied to the AW609 Tiltrotor Certification Basis
Authors	William Fraser, David King, Joseph M. Schaeffer, Dan Wells (Agusta Westland Philadelphia Corp.)
Date	Presented at the AHS International 74th Annual Forum & Technology Display, Phoenix, Arizona, USA, May 14-17, 2018. Copyright © 2018 by AHS International, Inc. All rights reserved.
Availability	https://vtol.org/store/product/development-of-poweredlift-airworthiness-standards-as-applied-to-the-aw609-tiltrotor-certification-basis-12876.cfm
Abstract	The primary purpose of the paper is to describe the safety considerations used to define the airworthiness requirements incorporated into the AW609 certification basis and how this forms the basis for airworthiness standards for a new powered-lift category of aircraft. The paper will provide a general history of and the process used to determine the certification basis, and provide specific examples of how the regulations reflect safety requirements. Examples of how the AW609 plans to meet the requirements are described. Specific examples given in this paper include, but are not limited to, definition and compliance to specific safety requirements (TR.1309), considerations for the unique flight control system including a nacelle conversion control device, pilot training requirements, ditching, fuel reserves, Transport Category Performance (ability to safely take off and land after an engine failure), All-Engines-Inoperative (AEI) operations, including autorotation. Each of these items will be expanded and described.

Title	FUELEAP Model-Based System Safety Analysis
Authors	Woodham et al.
Date	2018
Availability	https://arc.aiaa.org/doi/abs/10.2514/6.2018-3362
Abstract	NASA researchers, in a partnership with Boeing, are investigating a fuel-cell powered variant of the X-57 "Maxwell" Mod-II electric propulsion aircraft, which is itself derived from a stock Tecnam P2006T. The "Fostering Ultra-Efficient Low-Emitting Aviation Power" (FUELEAP) project will replace the X-57 power subsystem with a hybrid Solid-Oxide Fuel Cell (SOFC) system

	to increase the potential range of the electric-propulsion aircraft while dramatically improving efficiency and emissions over stock internal-combustion engines.

Title	VTOL Urban Air Mobility Concept Vehicles for Technology Development
Authors	Christopher Silva, Wayne Johnson (NASA Ames), Kevin Antcliff, Michael Patterson (NASA LaRC)
Date	2018
Availability	https://rotorcraft.arc.nasa.gov/Publications/files/vtol-urban-air-2.pdf
Abstract	The current push for Urban Air Mobility (UAM) is predicated on the feasibility of novel aircraft types, which will be enabled by the near-term availability of mature technology for high performance subsystems. A number of candidate concept aircraft are presently being designed to meet a set of UAM requirements, in order to quantify the tradeoffs and performance targets necessary for practical implementation of the UAM vision. In examining these vehicles, performance targets and recurring technology themes emerge, which may guide investments in research and development within NASA, other government agencies, academia, and industry.

Title	Concept Vehicles for VTOL Air Taxi Operations
Authors	Wayne Johnson, Christopher Silva, Eduardo Solis (NASA Ames Research Center)
Date	Presented at the AHS Technical Conference on Aeromechanics Design for Transformative Vertical Flight, San Francisco, CA, January 16-19, 2018
Availability	https://rotorcraft.arc.nasa.gov/Publications/files/Johnson_2018_TechMx.pdf
Abstract	Concept vehicles are presented for air taxi operations, also known as urban air mobility or on-demand mobility applications. Considering the design-space dimensions of payload (passengers and pilot), range, aircraft type, and propulsion system, three aircraft are designed: a single-passenger (250-lb payload), 50-nm range quadrotor with electric propulsion; a six-passenger (1200-lb payload), 4x50 = 200-nm range side-by-side helicopter with hybrid propulsion; and a fifteen-passenger (3000-lb payload), 8x50 = 400-nm range tilt-wing with turbo-electric propulsion. These concept vehicles are intended to focus and guide NASA research activities in support of aircraft development for emerging aviation markets, in particular VTOL air taxi operations. Research areas are discussed, illustrated by results from the design of the concept vehicles.

Title	Improving Safety by Reducing Design Assurance Overhead - Presentation
Authors	Garmin
Date	April 8, 2015
Availability	https://www.aea.net/events/rotorcraft/files/Apr2015/Garmin.pdf
Abstract	Discussion Topics •Design Assurance Considerations for Change •Proportionate Design Assurance •Certification Uncertainty Considerations for Change •Conclusion and Recommendations

Title	Regulatory Compliance in Multi-Tier Supplier Networks
Authors	Emray R. Goossen and Duke A. Buster, Honeywell International, Albuquerque, NM
Date	November 2014
Availability	https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150001255.pdf
Abstract	Over the years, avionics systems have increased in complexity to the point where 1st tier suppliers to an aircraft OEM find it financially beneficial to outsource designs of subsystems to 2nd tier and at times to 3rd tier suppliers. Combined with challenging schedule and budgetary pressures, the environment in which safety-critical systems are being developed introduces new hurdles for regulatory agencies and industry. This new environment of both complex systems and tiered development has raised concerns in the ability of the designers to ensure safety considerations are fully addressed throughout the tier levels. This has also raised questions about the sufficiency of current regulatory guidance to ensure: proper flow down of safety awareness, avionics application understanding at the lower tiers, OEM and 1st tier oversight practices, and capabilities of lower tier suppliers. Therefore, NASA established a research project to address Regulatory Compliance in a Multi-tier Supplier Network. This research was divided into three major study efforts: 1. Describe Modern Multi-tier Avionics Development 2. Identify Current Issues in Achieving Safety and Regulatory Compliance 3. Short-term/Long-term Recommendations Toward Higher Assurance Confidence This report presents our findings of the risks, weaknesses, and our recommendations. It also includes a collection of industry-identified risks, an assessment of guideline weaknesses related to multi-tier development of complex avionics systems, and a postulation of

	potential modifications to guidelines to close the identified risks and weaknesses.

Title	Software Fault Tolerance: A Tutorial
Authors	Wilfredo Torres-Pomales, NASA Langley Research Center
Date	October 2000
Availability	https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000120144.pdf
Abstract	Because of our present inability to produce error-free software, software fault tolerance is and will continue to be an important consideration in software systems. The root cause of software design errors is the complexity of the systems. Compounding the problems in building correct software is the difficulty in assessing the correctness of software for highly complex systems. This paper presents a review of software fault tolerance. After a brief overview of the software development processes, we note how hard-to-detect design faults are likely to be introduced during development and how software faults tend to be state-dependent and activated by particular input sequences. Although component reliability is an important quality measure for system level analysis, software reliability is hard to characterize and the use of post-verification reliability estimates remains a controversial issue. For some applications software safety is more important than reliability, and fault tolerance techniques used in those applications are aimed at preventing catastrophes. Single version software fault tolerance techniques discussed include system structuring and closure, atomic actions, inline fault detection, exception handling, and others. Multiversion techniques are based on the assumption that software built differently should fail differently and thus, if one of the redundant versions fails, at least one of the others should provide an acceptable output. Recovery blocks, Nversion programming, N self-checking programming, consensus recovery blocks, and t/(n-1) techniques are reviewed. Current research in software engineering focuses on establishing patterns in the software structure and trying to understand the practice of software engineering. It is expected that software fault tolerance research will benefit from this research by enabling greater predictability of the dependability of software.

Title	Certification Authorities Software Team (CAST) Position Paper CAST-24 Reliance on Development Assurance Alone when Performing a Complex and Full-Time Critical Function
Authors	Certification Authorities Software Team (CAST)
Date	March 2006

Availability	https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast_papers/media/cast-24.pdf
Abstract	Paper Purpose It is recognized today that in designing aircraft systems, manufacturers should prevent any single failure that leads to a catastrophic failure condition (JAR/FAR 25.1309 (extremely improbable); AMJ/AC 25.1309-1A). The fail-safe concept and techniques are discussed in the AMJ/AC 25.1309-1A to support this approach. [Single failures leading to a catastrophic event are prevented (occurrence extremely improbable) by the FAR/JAR, as well as multiple failures (25.1309 (d)(2))] However, when the failure is caused by a development error in the system, particularly in software or complex electronic hardware, the guidance materials are not clear on the applicability of fail-safe concept and techniques. Thus, the applicant and system designers need to consider the potential effect of such errors in the aircraft-level safety assessment, in order to ensure that their proposed system design and implementation of complex, safety-related systems can be demonstrated to have achieved an acceptable level of safety. The purpose of this paper is to highlight that development assurance alone is not necessarily sufficient to establish an acceptable level of safety for complex and full-time critical functions implemented in software or complex hardware. The paper presents rationale for the use of mitigation means in the system development to prevent either software or complex electronic hardware development errors from becoming a common point of failure that could lead to an unacceptable safety event (accident or incident). NOTE: This position paper has been coordinated among the software specialists of certification authorities from North America, South America, and Europe. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.

Title	Heavy Lift Vehicle (HLV) Avionics Flight Computing Architecture Study
Authors	Hudson et al.
Date	August 1, 2011
Availability	https://naca.larc.nasa.gov/search.jsp?R=20110014793&qs=N%3D4294957355%2B4294950110%2B4294926580
Abstract	A NASA multi-Center study team was assembled from LaRC, MSFC, KSC, JSC and WFF to examine potential flight computing architectures for a Heavy Lift Vehicle (HLV) to better understand avionics drivers. The study examined Design Reference Missions (DRMs) and vehicle requirements that could impact the vehicles avionics. The study considered multiple self-checking and voting architectural variants and examined reliability, fault-tolerance, mass,

	power, and redundancy management impacts. Furthermore, a goal of the study was to develop the skills and tools needed to rapidly assess additional architectures should requirements or assumptions change.

Title	AW609
Authors	Misc.
Date	2012
Availability	Public
Abstract	The AgustaWestland AW609 is a tilt-rotor VTOL aircraft aimed at the civilian market with expected FAA certification in 2019. The Wikipedia page states: "In 2012, the Federal Aviation Administration (FAA) stated that the AW609 was to be certified in compliance with both helicopter and fixed-wing aircraft rules; additionally, new codes were to be developed to cover the transition phase between the two modes." Related webpages: https://www.leonardocompany.com/en/product-services/elicotteri_helicopters/aw609 https://en.wikipedia.org/wiki/AgustaWestland_AW609

Title	FAA Safety Continuum Doctrine
Authors	Standards Management Team, Aircraft Certification Service (AIR)
Date	September 2014
Availability	https://www.regulations.gov/document?D=FAA-2015-1621-0018
Abstract	Beginning in 2012, the Aircraft Certification Service (AIR) began a conscious effort to increase awareness of the safety continuum amongst all Aircraft Certification employees. An increase in awareness is critical to achieving the next level of product safety. The safety continuum is a fundamental element of the AIR: 2018 Vision.

Title	Safety Considerations in Emerging Electric Aircraft Architectures
Authors	Christopher Courtin, R. John Hansman (MIT International Center for Air Transportation)
Date	October 11, 2018
Availability	http://hdl.handle.net/1721.1/118438
Abstract	Safety and certification considerations which impact the design of an emerging new class of small, electric aircraft were investigated. Based on an assessment of the different emerging aircraft designs, vehicles were grouped based on lifting and propulsive architecture. Likely certification pathways and the associated airworthiness requirements were investigated. Key hazards were identified, and were classified by severity for each

	architecture group. The key hazards identified were lithium-polymer battery thermal runaway and energy uncertainty, common mode power system failure, and vehicle automation failure. Mitigation strategies for each identified hazard were identified based on current technology and regulatory requirements. These mitigation strategies were assessed for different vehicle architectures. Aircraft with the ability to controllably glide or autorotate are shown to have lower certification risk.

Title	An Integrated Approach to Evaluating Risk Mitigation Measures for UAV Operational Concepts in the NAS
Authors	Roland E. Weibel, R. John Hansman, Jr. (MIT)
Date	September 2005 (AIAA Infotech@Aerospace Conference)
Availability	http://hdl.handle.net/1721.1/34907
Abstract	An integrated approach is outlined in this paper to evaluate risks posed by operating Unmanned Aerial Vehicles in the National Airspace System. The approach supports the systematic evaluation of potential risk mitigation measures recognizing key issues in creation of regulatory and safety policy, including public perception and UAV market forces. Risk mitigation measures are examined for two example concepts of operation: High Altitude Long Endurance UAV and small, local UAV operations. Primary hazards of ground impact and midair collision are considered. The examples illustrate three major areas of risk mitigation: exposure, recovery, and effects mitigation. The different mitigation possibilities raise key issues on how to determine appropriate UAV policies to ensure that an acceptable level of safety is achieved.

Title	Safety Considerations for Operation of Unmanned Aerial Vehicles in the National Airspace System
Authors	Roland E. Weibel, R. John Hnasman (MIT)
Date	November 21, 2006
Availability	http://hdl.handle.net/1721.1/34912
Abstract	There is currently a broad effort underway in the United States and internationally by several organizations to craft regulations enabling the safe operation of UAVs in the NAS. Current federal regulations governing unmanned aircraft are limited in scope, and the lack of regulations is a barrier to achieving the full potential benefit of UAV operations. To inform future FAA regulations, an investigation of the safety considerations for UAV operation in the NAS was performed. Key issues relevant to operations in the NAS, including performance and operating architecture were examined, as well as current rules and regulations governing unmanned aircraft. In integrating UAV operations in the NAS, it will be important to consider the implications of different levels of vehicle control and autonomous capability and the source of traffic surveillance in the system. A system safety analysis was performed according to FAA system safety guidelines for two critical

	hazards in UAV operation: midair collision and ground impact. Event-based models were developed describing the likelihood of ground fatalities and midair collisions under several assumptions. From the models, a risk analysis was performed calculating the expected level of safety for each hazard without mitigation. The variation of expected level of safety was determined based on vehicle characteristics and population density for the ground impact hazard, and traffic density for midair collisions. The results of the safety analysis indicate that it may be possible to operate small UAVs with few operational and size restrictions over the majority of the United States. As UAV mass increases, mitigation measures must be utilized to further reduce both ground impact and midair collision risks to target levels from FAA guidance. It is in the public interest to achieve the full benefits of UAV operations, while still preserving safety through effective mitigation of risks with the least possible restrictions. Therefore, a framework was presented under which several potential mitigation measures were introduced and could be evaluated. It is likely that UAVs will be significant users of the future NAS, and this report provides an analytical basis for evaluating future regulatory decisions.

Title	Definition of an airworthiness certification framework for civil unmanned aircraft systems
Authors	Clothier, Reece A., Palmer, Jennifer L., Walker, Rodney A.,& Fulton, Neale L.
Date	2011
Availability	https://www.researchgate.net/publication/251614897_Definition_of_an_airworthiness_certification_framework_for_civil_unmanned_aircraft_systems
Abstract	The development of effective safety regulations for unmanned aircraft systems (UAS) is an issue of paramount concern for industry. The development of this framework is a prerequisite for greater UAS access to civil airspace and, subsequently, the continued growth of the UAS industry. The direct use of the existing conventionally piloted aircraft (CPA) airworthiness certification framework for the regulation of UAS has a number of limitations. The objective of this paper is to present one possible approach for the structuring of airworthiness regulations for civilian UAS. The proposed approach facilitates a more systematic, objective and justifiable method for managing the spectrum of risk associated with the diversity of UAS and their potential operations. A risk matrix is used to guide the development of an airworthiness certification matrix (ACM). The ACM provides a structured categorisation that facilitates the future tailoring of regulations proportionate to the levels of risk associated with the operation of the UAS. As a result, an objective and traceable link may be established between mandated regulations and the overarching objective for an equivalent level of safety to CPA. The ACM also facilitates the systematic consideration of a range of technical and operational mitigation strategies. For these reasons, the ACM is proposed as a suitable method for the structuring of an airworthiness certification framework for civil or commercially operated UAS (i.e., the UAS equivalent in function to the Part 21 regulations for civil CPA) and for the

	further structuring of requirements on the operation of UAS in un-segregated airspace.

Title	Executive Briefing Urban Air Mobility (UAM) Market Study
Authors	Booz Allen Hamilton
Date	October 5, 2018
Availability	***Not for public release.*** https://www.nasa.gov/sites/default/files/atoms/files/bah_uam_executive_briefing_181005_tagged.pdf
Abstract	Our analysis focused on three potential UAM markets: Airport Shuttle, Air Taxi, and Air Ambulance using ten target urban areas to explore market size and barriers.

Title	Part 23 Accepted Means of Compliance based on ASTM Consensus Standards
Authors	FAA
Date	May 11, 2018
Availability	https://www.faa.gov/aircraft/air_cert/design_approvals/small_airplanes/small_airplanes_regs/
Abstract	N/A

Title	ASTM F3269-17, Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions
Authors	ASTM International, Subcommittee F38.01 on Airworthiness
Date	2017
Availability	Available for purchase. https://www.astm.org/Standards/F3269.htm
Abstract	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.

Title	ASTM International Committee F38 on Unmanned Aircraft Systems ASTM Meeting at AUVSI, Dallas, TX
Authors	Ted Wierzbanowski, Chair, ASTM International Committee F38
Date	May 8, 2017

Availability	https://www.astm.org/COMMIT/2017-AUVSI%20May%20Meeting.pdf
Abstract	ASTM Meeting at AUVSI, Dallas, TX Vision, Mission, & Structure Focus on small UAS (sUAS/sRPAS) – Background, Published standards , Other sUAS standards in development Global Acceptance of sUAS Standards Conclusion

Title	Revision of F3269 - 17 Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions
Authors	ASTM International, Subcommittee F38.01 on Airworthiness
Date	9/7/2018
Availability	https://www.astm.org/DATABASE.CART/WORKITEMS/WK65056.htm
Abstract	Goal is to develop the standard to a level of capability that defines run-time monitoring (RTA) attributes to a level that the FAA or CAA will agree that monitors developed to this standard are sufficient to allow the UAS to evolve the complex function with its associated avionics equipment and sensors without requiring vehicle recertification as the CONOPS evolve after initial certification. a. Provide additional guidance on Safety Monitor design best practices, to explicitly include guidance on partitioning, dissimilarity, and the option for multiple individual safety monitors comprising the Safety Monitor function, as well as defining safety monitor classes and key attributes. b. Provide additional use cases as Appendices. c. Provide additional information contrasting the F3269 approach with other architectural approaches (e.g., SAE ARP 4754A, RTCA DO-178C). d. Modify requirements to performance based to allow multiple implementation and implementation architectures e. Make additional updates as required.

Title	Initial considerations of a multi-layered run time assurance approach to enable unpiloted aircraft
Authors	L. R. Hook, M. Skoog, M. Garland, W. Ryan, D. Sizoo and J. VanHoudt
Date	2018 IEEE Aerospace Conference, Big Sky, MT, 2018
Availability	https://ieeexplore.ieee.org/document/8396622
Abstract	Increased autonomy promises many advantages in the aviation domain, especially for unmanned aerial systems and small aircraft. However, several critical challenges remain and must be solved before highly autonomous or “unpiloted” operation can be fully realized. Of these challenges, safety and safety assurance is of utmost importance. Run-time assurance (RTA) has been shown, both theoretically and experimentally, to be a very promising avenue upon which to assure safety of the myriad of functions required for effective flight. This paper develops concepts based around an RTA architecture composed of multiple safety monitors encompassing and

	assuring many flight functions. Implementation of the concepts described in this work was performed on a small-unmanned air vehicle (SUAV) as a part of the NASA/FAA Traveler project. Lessons learned in requirements generation, system design, and operations are presented along with flight test results. These results are applicable to not only SUAV, but also to small manned aircraft and other vehicle systems where safety assured autonomy is desired.

Title	Certification strategies using run-time safety assurance for part 23 autopilot systems
Authors	L.R. Hook, M. Clark, D. Sizoo, M.A. Skoog, J. Brady
Date	Aerospace Conference 2016 IEEE
Availability	https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170007254.pdf
Abstract	Part 23 aircraft operation, and in particular general aviation, is relatively unsafe when compared to other common forms of vehicle travel. Currently, there exists technologies that could increase safety statistics for these aircraft; however, the high burden and cost of performing the requisite safety critical certification processes for these systems limits their proliferation. For this reason, many entities, including the Federal Aviation Administration, NASA, and the US Air Force, are considering new options for certification for technologies which will improve aircraft safety. Of particular interest, are low cost autopilot systems for general aviation aircraft, as these systems have the potential to positively and significantly affect safety statistics. This paper proposes new systems and techniques, leveraging run-time verification, for the assurance of general aviation autopilot systems, which would be used to supplement the current certification process and provide a viable path for near-term low-cost implementation. In addition, discussions on preliminary experimentation and building the assurance case for a system, based on these principles, is provided.

Title	An ASTM Standard for Bounding Behavior of Adaptive Algorithms for Unmanned Aircraft Operations (Invited)
Authors	Stephen P. Cook
Date	AIAA Information Systems-AIAA Infotech @ Aerospace, AIAA SciTech Forum, (AIAA 2017-0881)
Availability	https://arc.aiaa.org/doi/abs/10.2514/6.2017-0881 http://mys5.org/Proceedings/2017/Day_1/2017-S5-Day1_1405_Cook.pdf

Abstract	The integration of intelligent systems in aerospace systems offers enormous opportunities to improve the safety and performance of military and civil aircraft. However, challenges associated with the certification of aircraft containing complex functions and adaptive algorithms remain a barrier to realizing the full potential of these technologies. Industry consensus standards provide a way for airworthiness authorities to establish acceptable means of compliance to ensure safety. Recently ASTM F38 Committee on Unmanned Aircraft Systems embarked on the task of developing a standard practice to bound the flight behavior of unmanned aircraft systems containing complex functions. This standard practice defines requirements for a run-time assurance architecture that ensure the flight behavior of an unmanned aircraft system containing complex functions is safely bounded. This paper will discuss the philosophy behind the standard, its development, and the components of a generic run-time assurance architecture.

Title	Improving Safety by Reducing Design Assurance Overhead
Authors	Garmin
Date	April 8, 2015
Availability	Presented to AEA / GAMA Rotorcraft Forum https://www.aea.net/events/rotorcraft/files/Apr2015/Garmin.pdf
Abstract	Design Assurance Considerations for Change Proportionate Design Assurance Certification Uncertainty Considerations for Change Conclusion and Recommendations

Title	SOFTWARE ASSURANCE APPROACHES, CONSIDERATIONS, AND LIMITATIONS FINAL REPORT
Authors	Mats Heimdahl, University of Minnesota; Nancy Leveson, Massachusetts Institute of Technology. Julie Redler, Melanie Felton, and Grady Lee are from Safeware Engineering Corporation
Date	October 2016
Availability	Report Number: DOT/FAA/TC-15/57 This document is available to the U.S. public through the National Technical Information Services (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.
Abstract	The cost of developing software in compliance with RTCA/DO-178B/RTCA/DO-278 is generally high. Nevertheless, these standards have helped to ensure the development of software systems of high integrity with excellent operational histories. The "Alternative Approaches to Software Assurance" three-phase study was undertaken to evaluate the current state of software assurance processes and propose alternative approaches with the potential to streamline the process and reduce the assurance costs

	without compromising safety. Phase 1 work focused on three areas: an examination of alternative methods, a comparison of aerospace industry standards to other safety-critical industry's standards, and a poll to query aviation industry personnel on their experience with DO-178B and DO-278. The findings from Phase 1 did not highlight any alternative approaches that could replace DO-178B or DO-278. The authors recommended looking at technical advances that could still meet the goal of the study but were not necessarily alternatives to DO-178B and DO-278. The Phase 1 findings directed the team to look at techniques that could help users of the standards to streamline the process (and realize cost benefits) by ensuring the requirements were complete and correct early in the development process. The goal of Phase 2 was to conduct an in-depth study of techniques that warranted further study from Phase 1, including: hazard analysis; human reviews; model-based specification and analysis; architectural modeling and analysis; and collection of information regarding how each approach helps in streamlining the certification process and which approaches are best used for commercial off-the-shelf and legacy software. The research from the first two phases directed the team to further focus on Systems Theoretic Process Analysis (STPA), model-based development, and formal verification in the third phase. Although these methods have been around for some time, there have been advancements in model-based development and formal verification that deemed it worthwhile to re-visit them. The Phase 3 work also highlighted how STPA can catch more system and software errors in the requirements than the traditional hazard analysis techniques, such as fault tree analysis. The analysis demonstrated how STPA could be applied to a flight guidance system and how hazard causes could be mitigated. The research also looked at cost savings that were realized by Rockwell Collins when they used model-based development and by Airbus when they used formal verification on their projects. A discussion about the pitfalls of using model-based development and formal verification was also included.

Title	Overarching Properties
Authors	Michael Holloway, NASA LaRC
Date	11/14/2018
Availability	***Not for public release.*** Provided by Kurt Woodham on 11/29/2018.
Abstract	Overview of Overarching Properties

Title	Understanding the Overarching Properties
Authors	Michael Holloway, NASA LaRC
Date	11/28/2018
Availability	***Not for public release.*** Provide by Kurt Woodham on 11/29/2018.
Abstract	The purpose of this document is to explain the Overarching Properties, including their philosophical foundation, the specific details of each property,

	the relationships among them, and some practical considerations that attach to their use. Readers of this document are assumed to be at least somewhat familiar with current laws, regulations, and processes governing certification of airborne systems, software, and electronic hardware. Because the Overarching Properties are expressed at a much higher level of abstraction than is common today, however, readers without intimate knowledge of current practice may find understanding the Overarching Properties easier than readers with such knowledge.

Title	Understanding Assurance Cases: An Educational Series in Five Parts
Authors	Michael Holloway, NASA LaRC
Date	November 26, 2018
Availability	https://shemesh.larc.nasa.gov/arg/uac.html
Abstract	At the prompting of friends from the FAA, I recently converted to 508-compliant PDF the 5-part educational series (Understanding Assurance Cases) I did for them in 2015/6. The material has successfully completed approval through TPSAS.

Title	F3309- 18 Standard Practice for Simplified Safety Assessment of Systems and Equipment in Small Aircraft
Authors	Active Standard ASTM F3309 / F3309M | Developed by Subcommittee: F44.50
Date	2018
Availability	https://www.astm.org/Standards/F3309.htm
Abstract	This practice covers methods for conducting a simplified safety assessment of aircraft systems and equipment. The material was developed through open consensus of international experts in general aviation. This information was created by focusing on Level 1 and Level 2 Normal Category aeroplanes employing conventional systems. The content may be more broadly applicable. It is the responsibility of the Applicant to substantiate broader applicability as a specific means of compliance. If the criteria specified within this simplified practice is deemed not to be relevant to a particular application, the Applicant should use the safety assessment process defined in Practice F3230. The topics covered within this practice are: Procedural Flowchart, Failure Condition Identification and Classification, Safety Objectives, Design and Installation Appraisal, Qualitative Analysis of Failure Conditions, Common Mode Analysis, Use of Similarity, and Documentation.

Title	ASTM F3235 - 17a Standard Specification for Aircraft Storage Batteries
Authors	Active Standard ASTM F3235 | Developed by Subcommittee: F44.50
Date	2017
Availability	https://www.astm.org/Standards/F3235.htm

Abstract	This specification covers international standards for the electrical storage battery aspects of airworthiness and design for “small” aircraft.

Title	ASTM F3316 / F3316M - 18 Standard Specification for Electrical Systems for Aircraft with Electric or Hybrid-Electric Propulsion
Authors	Active Standard ASTM F3316 / F3316M | Developed by Subcommittee: F44.50
Date	2018
Availability	https://www.astm.org/Standards/F3316.htm
Abstract	This specification covers the electrical systems, electrical equipment, and electrical power distribution aspects of airworthiness and design for aircraft with Electric or Hybrid-Electric Propulsion. This material was developed through open consensus of international experts in general aviation. This material was created by focusing on Normal Category Airplanes. The content may be more broadly applicable; it is the responsibility of the applicant to substantiate broader applicability as a specific means of compliance.

Appendix E References

Ref. No **Reference Title**

[1] Title 14, Code of Federal Regulations, Part 1, Definitions and Abbreviations

[2] Title 14, Code of Federal Regulations, Part 21, Certification Procedures for Products and Parts

[3] Title 14, Code of Federal Regulations, Part 23 – Airworthiness Standards: Normal Category Airplanes, Amendment 23-64, December 2016

[4] Title 14, Code of Federal Regulations, Part 27 – Airworthiness Standards: Normal Category Rotorcraft

[5] AC 27-1B, Chg 8, Certification of Normal Category Rotorcraft, June 2018

[6] AC 23.1309-1E, System Safety Analysis and Assessment for Part 23 Airplanes, November 2011

[7] Advisory Circular (AC) 23.2010-1, FAA Accepted Means of Compliance Process for 14 Code of Federal Regulations (CFR) Part 23, March 2017

[8] PS-AIR-21.8-1602, Non-Required Safety Enhancing Equipment (NORSEE)

[9] PS-ASW-27-15, Safety Continuum for Part 27 Normal Category Rotorcraft Systems and Equipment, June 2017

[10] Title 14, Code of Federal Regulations, Part 29 - Airworthiness Standards: Transport Category Rotorcraft

[11] Special Condition, Vertical Take-Off and Landing (VTOL) Aircraft, SC-VTOL-01, 02-July-2019

[12] Overarching Properties, 2018-11-14, FAA-NASA Working Group

[13] Federal Register Volume 83, No 92 Notice 23-81-NOA

[14] PS-AIR-23-09 System Level Verification of Electronic Equipment (Software and Airborne Electronic Hardware)

[15] ASTM F3264-18, Standard Specification for Normal Category Aeroplanes Certification, March 2018

[16] ASTM F3061-17 Standard Specification for Systems and Equipment in Small Aircraft, March 2017

[17] ASTM F3230-17 Standard Practice for Safety Assessment of Systems and Equipment in Small Aircraft, March 2017

[18] ASTM F3153-15 Standard Specification for Verification of Avionics Systems, September 2015

[19] ASTM F3269-17 Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions

[20] ASTM F3309-18 Standard Practice for Simplified Safety Assessment of Systems and Equipment in Small Aircraft

[21] ASTM F3338-18, Standard Specification for Design of Electric Propulsion Units for General Aviation Aircraft

[22] SAE ARP 4754A, Guidelines for Development of Civil Aircraft and Systems, December 2010

[23] SAE ARP 4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, December 1996

[24] RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, December 1992

[25] RTCA DO-178C, Software Considerations in Airborne Systems and Equipment Certification, December 2011

[26] RTCA DO-254, Design Assurance Guidance for Airborne Electronic Hardware, April 2000

[27] RTCA DO-331, Model-Based Development and Verification Supplement to DO-178C and DO-278A, December 2011

[28] Mike DeWalt and G. Frank McCormick. “Technology Independent Assurance Method”, 978-1-4799-5001-0 IEEE, 33rd Digital Avionics Systems Conference, October 2014

[29] Lui Sha, Ragunathan Rajkumar, and Michael Gagliardl. A Software Architecture for Dependable and Evolvable Industrial Computing Systems. No. CMU/SEI-95-TR-005. Carnegie-Mellon University, Software Engineering Institute, 1995.

[30] Jose G. Rivera, Alejandro A. Danylyszyn, Charles B. Weinstock, Lui R. Sha, and Michael J. Gagliardi. An Architectural Description of the Simplex Architecture. No. CMU/SEI-96-TR-006. Carnegie-Mellon University, Software Engineering Institute, 1996.

[31] Danbing Seto, Bruce H. Krogh, Lui Sha, and A. Chutinan. "Dynamic control system upgrade using the simplex architecture." IEEE Control Systems Magazine 18, no. 4 (1998): 72-80.

[32] Danbing Seto, Bruce Krogh, Lui Sha, and Alongkrit Chutinan. "The Simplex architecture for safe online control system upgrades." In Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207), vol. 6, pp. 3504-3508. IEEE, 1998.

[33] Danbing Seto, and Lui Sha. "An Engineering Method for Safety Region Development." Carnegie Mellon Software Engineering Institute Technical Report CMU/SEI-99-TR-018, 1999.

[34] Tanya L. Crenshaw, Elsa Gunter, Craig L. Robinson, Lui Sha, and P. R. Kumar. "The simplex reference model: Limiting fault-propagation due to unreliable components in cyber-physical system architectures." In 28th IEEE International Real-Time Systems Symposium (RTSS 2007), pp. 400-412. IEEE, 2007.

[35] Stanley Bak, Deepti K. Chivukula, Olugbemiga Adekunle, Mu Sun, Marco Caccamo, and Lui Sha. "The system-level simplex architecture for improved real-time embedded system safety." In 2009 15th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 99-107. IEEE, 2009.

[36] Prasanth Vivekanandan, Gonzalo Garcia, Heechul Yun, and Shawn Keshmiri. "A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles." In 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pp. 69-75. IEEE, 2016.

[37] Stanley Bak, Fardin Abdi Taghi Abad, Zhenqi Huang, and Marco Caccamo. "Using run-time checking to provide safety and progress for distributed cyber-physical systems." In 2013 IEEE 19th International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 287-296. IEEE, 2013.

[38] Jianguo Yao, Xue Liu, Guchuan Zhu, and Lui Sha. "NetSimplex: Controller fault tolerance architecture in networked control systems." IEEE Transactions on Industrial Informatics 9, no. 1 (2013): 346-356.

[39] Xiaofeng Wang, Naira Hovakimyan, and Lui Sha. “L1Simplex: fault-tolerant control of cyber-physical systems.” In 2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), pp. 41-50. IEEE, 2013.

[40] Xiaofeng Wang, Naira Hovakimyan, and Lui Sha. “RSimplex: A Robust Control Architecture for CyberaAnd Physical Failures.” ACM Transactions on Cyber-Physical Systems 2, no. 4 (2018): 27.

[41] Alec Bateman, Carl Elks, David Ward, and John Schierman. "New verification and validation methods for guidance/control of advanced autonomous systems." In Infotech@ Aerospace, p. 7117. 2005.

[42] John Schierman, David Ward, Brian Dutoi, Anthony Aiello, John Berryman, Michael DeVore, Walter Storm, and Jason Wadley. "Run-time verification and validation for safety-critical flight control systems." In AIAA Guidance, Navigation and Control Conference and Exhibit, p. 6338. 2008.

[43] John D. Schierman, Michael D. DeVore, Nathan D. Richards, Neha Gandhi, Jared K. Cooper, Kenneth R. Horneman, Scott Stoller, and Scott Smolka. Runtime assurance framework development for highly adaptive flight control systems. Technical report AFRL-RQ-WP-TR-2016-0001, AFRL. 2016.

[44] John D. Schierman, Michael DeVore, Nathan D. Richards, and Matthew Clark. "The Introduction of Software Runtime Protection for Autonomous Aerospace Systems." In AIAA Information Systems-AIAA Infotech@ Aerospace. 2017.

[45] Loyd R. Hook, Matthew Clark, David Sizoo, Mark A. Skoog, and James Brady. "Certification strategies using run-time safety assurance for part 23 autopilot systems." In 2016 IEEE Aerospace Conference, pp. 1-10. IEEE, 2016.

[46] Justin G. Fuller, Loyd R. Hook, Nathan Hutchins, K. Niki Maleki, and Mark A. Skoog. "Toward run-time assurance in general aviation and unmanned aircraft vehicle autopilots." In 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), pp. 1-9. IEEE, 2016. DOI 10.1109/DASC.2016.7778100

[47] Justin G. Fuller, Loyd R. Hook, and Nathan Hutchins. "Accounting for helpful and harmful human reactions in run-time assurance frameworks." In 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), pp. 1-8. IEEE, 2017. DOI 10.1109/DASC.2017.8102057

[48] Hook, Loyd R., Mark Skoog, Michael Garland, Wes Ryan, Dave Sizoo, and John VanHoudt. "Initial considerations of a multi-layered run time assurance approach to enable unpiloted aircraft." In 2018 IEEE Aerospace Conference, pp. 1-11. IEEE, 2018.

[49] Mark Skoog. “Verifying a Highly Autonomous Unmanned Aircraft.” Available at https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170004610.pdf. NASA, 2017

[50] Modern Technology Solutions, Inc. “NASA and MTSI to Partner towards Developing a Certifiable Autonomous Aircraft Framework,” available at https://www.mtsi-va.com/nasa-and-mtsi-to-partner-towards-developing-a-certifiable-autonomous-aircraft-framework/, 2018.

[51] Mark Skoog. “Trustworthy Autonomy Development and Flight Demonstration: Multi-Monitor Run Time Assurance Research Update” available at https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20180002830.pdf, NASA, 2018.

[52] Stefan Mitsch and André Platzer. "ModelPlex: Verified runtime validation of verified cyber-physical system models." Formal Methods in System Design 49, no. 1-2 (2016): 33-74.

[53] André Platzer. "Verified Models & Verified Runtime Validation." In Logical Foundations of Cyber-Physical Systems, pp. 557-575. Springer, Cham, 2018.

[54] Nathan Fulton and André Platzer. "Safe reinforcement learning via formal methods: Toward safe control through proof and learning." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[55] Kerianne H. Gross, Matthew A. Clark, Jonathan A. Hoffman, Eric D. Swenson, and Aaron W. Fifarek. "Run-time assurance and formal methods analysis nonlinear system applied to nonlinear system control." Journal of Aerospace Information Systems 14, no. 4. 2017.

[56] Remus Avram, Xiaodong Zhang, Jonathan A. Muse, and Matthew Clark. "Nonlinear Adaptive Control of Quadrotor UAVs with Run-Time Safety Assurance." In AIAA Guidance, Navigation, and Control Conference, p. 1896. 2017.

[57] Matthew Dillsaver, Matthew Clark, and Xiaodong Zhang. "Military Airworthiness Certification of Autonomous Air Vehicles with Adaptive Controllers." In AIAA Information Systems-AIAA Infotech@ Aerospace, p. 0564. 2017.

[58] Dung Phan, Junxing Yang, Matthew Clark, Radu Grosu, John Schierman, Scott Smolka, and Scott Stoller. "A component-based simplex architecture for high-assurance cyber-physical systems." In 2017 17th International Conference on Application of Concurrency to System Design (ACSD), pp. 49-58. IEEE, 2017.

[59] John D. Schierman, David Neal, Edmond Wong, and Amy K. Chicatelli. "Runtime Assurance Protection for Advanced Turbofan Engine Control." In 2018 AIAA Guidance, Navigation, and Control Conference. 2018.

[60] Alec J. Bateman, William Gressick, and Neha Gandhi. "Application of Run-time Assurance Architecture to Robust Geofencing of SUAS." In 2018 AIAA Information Systems-AIAA Infotech@ Aerospace. 2018.

[61] Lael Rudd. "Switch Control Architecture for Advanced Control System Certification." In AIAA Guidance, Navigation, and Control Conference, p. 5674. 2009.

[62] Stanley Bak, Ashley Greer, and Sayan Mitra. "Hybrid Cyberphysical System Verification with Simplex Using Discrete Abstractions." In 16th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2010), pp. 143-152. IEEE, 2010.

[63] Bak, Stanley, Karthik Manamcheri, Sayan Mitra, and Marco Caccamo. "Sandboxing controllers for cyber-physical systems." In 2011 IEEE/ACM Second International Conference on Cyber-Physical Systems, pp. 3-12. IEEE, 2011.

[64] Bak, Stanley, Taylor T. Johnson, Marco Caccamo, and Lui Sha. "Real-time reachability for verified simplex design." In 2014 IEEE Real-Time Systems Symposium, pp. 138-148. IEEE, 2014.

[65] Johnson, Taylor T., Stanley Bak, Marco Caccamo, and Lui Sha. "Real-time reachability for verified simplex design." ACM Transactions on Embedded Computing Systems (TECS) 15, no. 2 (2016): 26.

[66] Yang, Junxing, Md Ariful Islam, Abhishek Murthy, Scott A. Smolka, and Scott D. Stoller. "A Simplex Architecture for Hybrid Systems Using Barrier Certificates." In International Conference on Computer Safety, Reliability, and Security, pp. 117-131. Springer, Cham, 2017. DOI 10.1007/978-3-319-66266-4_8.

[67] **]** Zhang, Xiaodong, Matthew Clark, Kudip Rattan, and Jonathan Muse. "Controller verification in adaptive learning systems towards trusted autonomy." In Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, pp. 31-40. ACM, 2015.

[68] Ankush Desai, Shaz Qadeer, and Sanjit A. Seshia. "Programming Safe Robotics Systems: Challenges and Advances." In International Symposium on Leveraging Applications of Formal Methods, pp. 103-119. Springer, Cham, 2018.

[69] Ankush Desai, Shromona Ghosh, Sanjit A. Seshia, Natarajan Shankar, and Ashish Tiwari. "SOTER: programming safe robotics system using runtime assurance." arXiv preprint arXiv:1808.07921 (2018).

[70] Nelly Delgado, Ann Q. Gates, and Steve Roach. "A taxonomy and catalog of runtime software-fault monitoring tools." IEEE Transactions on software Engineering 30, no. 12 (2004): 859-872.

[71] Leucker, Martin, and Christian Schallhart. "A brief account of runtime verification." The Journal of Logic and Algebraic Programming 78, no. 5 (2009): 293-303.

[72] Falcone, Ylies, Klaus Havelund, and Giles Reger. "A Tutorial on Runtime Verification." Engineering dependable software systems 34 (2013): 141-175.

[73] Bartocci, Ezio, Yliès Falcone, Adrian Francalanza, and Giles Reger. "Introduction to runtime verification." In Lectures on Runtime Verification, pp. 1-33. Springer, Cham, 2018.

[74] Wilfredo Torres-Pomales. Software Fault Tolerance – A Tutorial. Technical report TR-2000-210616, NASA, 2000.

[75] Sha, Lui, John B. Goodenough, and Bill Pollak. "Simplex architecture: Meeting the challenges of using COTS in high-reliability systems." Crosstalk (1998): 7-10.

[76] John C. Knight and Nancy G. Leveson. "An experimental evaluation of the assumption of independence in multiversion programming." IEEE Transactions on software engineering 1 (1986): 96-109.

[77] Sha, Lui. "Using simplicity to control complexity." IEEE Software 4 (2001): 20-28.

[78] Knight, John C., and Elisabeth A. Strunk. "Achieving critical system survivability through software architectures." In Architecting Dependable Systems II, pp. 51-78. Springer, Berlin, Heidelberg, 2004.

[79] J. E. Wilborn and J. V. Foster, “Defining Commercial Transport Loss-of-Control: A Quantitative Approach,” in Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit 16 – 19 August 2004, Providence, Rhode Island, Aug. 2004.

[80] C. M. Belcastro and J. V. Foster, “Aircraft loss-of-control accident analysis,” in Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, 2010.

[81] Belcastro, Christine, and Steven Jacobson. "Future integrated systems concept for preventing aircraft loss-of-control accidents." In AIAA Guidance, Navigation, and Control Conference, p. 8142. 2010. DOI 10.2514/6.2010-8142.

[82] C. Belcastro, J. Foster, R. Newman, L. Groff, D. Crider, D. Klyde, and A. Huston, “Preliminary analysis of aircraft loss of control accidents: Worst case precursor combinations and temporal sequencing,” in Proceedings of the AIAA SciTech Guidance, Navigation and Control Conference, AIAA-2014-0612, 2014.

[83] Boeing Commercial Airplanes. Statistical summary of commercial jet airplane accidents. Worldwide Operations 1959 – 2008.

[84] A. Lambregts, G. Nesemeier, R. Newman, and J.Wilborn, “Airplane upsets: Old problem, new issues,” in AIAA Modeling and Simulation Technologies Conference and Exhibit, p. 6867, 2008.

[85] Les Dorr. Fact Sheet – General Aviation Safety. Available at https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=21274. FAA. 2018.

[86] C. M. Belcastro, R. L. Newman, J. Evans, D. H. Klyde, L. C. Barr, and E. Ancel, “Hazards identification and analysis for unmanned aircraft system operations,” in 17th AIAA Aviation Technology, Integration, and Operations Conference, p. 3269, 2017.

[87] Neha Gandhi, Nathan Richards, and Alec Bateman. "Desktop simulator demonstration of a joint human/automated upset recovery system." In AIAA Guidance, Navigation, and Control Conference, p. 4820. 2012.

[88] Nathan D. Richards, Neha Gandhi, Alec J. Bateman, David H. Klyde, and Amanda K. Lampton. "Development and Pilot-in-the-Loop Evaluation of Robust Upset-Recovery Guidance." In AIAA Guidance, Navigation, and Control Conference, p. 0879. 2016.

[89] Nathan D. Richards, Neha Gandhi, Alec J. Bateman, David H. Klyde, and Amanda K. Lampton. "Vehicle upset detection and recovery for onboard guidance and control." Journal of Guidance, Control, and Dynamics 40, no. 4 (2016): 920-933.

[90] Nathan D. Richards, Neha Gandhi, and Alec Bateman. "Improved upset recovery strategies through explicit consideration of pilot dynamic behavior." In AIAA Guidance, Navigation, and Control Conference, p. 4821. 2012.

[91] Small Airplane Issues List at https://www.faa.gov/aircraft/air_cert/design_approvals/small_airplanes/small_airplanes_regs/media/SAIL_FY19_Q2.pdf

[92] Stephen Jacklin, Johann Schumann, Pramod Gupta, M. Lowry, John Bosworth, Eddie Zavala, Kelly Hayhurst, Celeste Belcastro, and Christine Belcastro. "Verification, validation, and certification challenges for adaptive flight-critical control system software." In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 5258. 2004.

[93] Chris Wilkinson, Jonathan Lynch, and Raj Bharadwaj. *Regulatory considerations for adaptive systems.* Technical Report NASA/CR-2013-218010, NASA, June 2013.

[94] Chris Wilkinson, Jonathan Lynch, Raj Bharadwaj, and Kurt Woodham. *Verification of Adaptive Systems.* Technical Report DOT/FAA/TC-16/4. Federal Aviation Administration, April 2016.

[95] S. Bhattacharyya, D. Cofer, D. J. Musliner, J. Mueller, and E. Engstrom. *Certification considerations for adaptive systems*. Technical report CR-2015-218702, NASA, 2015.

[96] Alwyn Goodloe. "Challenges in high-assurance runtime verification." In *International Symposium on Leveraging Applications of Formal Methods*, pp. 446-460. Springer, Cham, 2016.

[97] Pike, Lee, Sebastian Niller, and Nis Wegmann. "Runtime verification for ultra-critical systems." In International Conference on Runtime Verification, pp. 310-324. Springer, Berlin, Heidelberg, 2011

[98] ASTM International. *Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions*. Active Standard ASTM F3269, 2017.

[99] Stephen P. Cook. "An ASTM Standard for Bounding Behavior of Adaptive Algorithms for Unmanned Aircraft Operations." In AIAA Information Systems-AIAA Infotech@ Aerospace, p. 0881. 2017.

[100] Clark, Matthew, Xenofon Koutsoukos, Ratnesh Kumar, Insup Lee, George Pappas, Lee Pike, Joseph Porter, and Oleg Sokolsky, “A Study on run time assurance for complex cyber physical systems,” Technical Report, Air Force Research Lab, Wright- Patterson AFB, 2013.

[101] Norbert Wiener. "The homogeneous chaos." American Journal of Mathematics 60, no. 4 (1938): 897-936.

[102] The Assurance Case Working Group. Goal Structuring Notation Community Standard Version 2, SCSC-141B. 2018.

[103] David J. Rinehart, John C. Knight, and Jonathan Rowanhill. "Current practices in constructing and evaluating assurance cases with applications to aviation." Technical Report NASA/CR–2015-218678, NASA. 2015.

[104] John Rushby. "How Do We Certify For The Unexpected?" In AIAA Guidance, Navigation and Control Conference and Exhibit, p. 6799. 2008.

[105] P. Perfect, M. D. White, and M. Jump, “Towards handling qualities requirements for future personal aerial vehicles,” in 69th Annual Forum of the American Helicopter Society, AHS, May 2013

[106] Slotine, Jean-Jacques E., and Weiping Li. *Applied nonlinear control*. Vol. 199. No. 1. Englewood Cliffs, NJ: Prentice hall, 1991

[107] Enns, Dale, et al. "Dynamic inversion: an evolving methodology for flight control design." International Journal of control 59.1 (1994): 71-91.

[108] Harris, Jeffrey J. "F-35 Flight Control Law Design, Development and Verification." 2018 Aviation Technology, Integration, and Operations Conference. 2018.

[109] Horn, Joseph F. "Non-Linear Dynamic Inversion Control Design for Rotorcraft." Aerospace 6.3 (2019): 38.

[110] Cooper, J., Schierman, J., and Horn, J. F., “Robust adaptive disturbance compensation for ship-based rotorcraft,” in AIAA Guidance, Navigation, and Control Conference, AIAA, 2010.

[111] ADS-33E-PRF, US Army Aviation and Missile Command, “Handling Qualities Requirements for Military Rotorcraft, 2000

[112] MIL-STD-1797A, Department of Defense Handbook, “Flying Qualities of Piloted Aircraft”, 1997

[113] L. E. Kavraki, S. P., L. J. C., and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in IEEE Trans. Robot. and Autom

[114] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning," in IEEE International Conference on Robotics and Automation

[115] T. Schouwenaars, B. DeMoor, E. Feron, and J. How, “Mixed integer programming for multi-vehicle path planning," in Proc. of the European Control Conference, Sep 2001

[116] N. Richards and R. Bird, “UAV 2D and 3D path planning for sensor-on-target maneuvers and obstacle avoidance." Barron Associates Technical Report 294. prepared for Northrop Grumman Software Enabled Control

[117] E. Frazzoli, M. A. Dahleh, and E. Feron, “Real-time motion planning for agile autonomous vehicles," JCD, vol. 25, pp. 116{129, January-February 2002

[118] Y. Wang, E. Akuiyibo, and S. Boyd, “Applications of convex optimization in control,” Talk, 2011. [Online]. Available: http://www.stanford.edu/~yw224/eth_talk.pdf

[119] Johnson, Wayne. *Rotorcraft aeromechanics*. Vol. 36. Cambridge University Press, 2013

[120] Prouty, R. W. *Helicopter aerodynamics*. PJS Publ., 1984

[121] Johnson, Wayne. "Model for vortex ring state influence on rotorcraft flight dynamics." (2005)

[122] ARAC SDAHWG, System Design and Analysis, Draft Advisory Circular/Advisory Material Joint, Arsenal, 2002

[123] MIL-HDBK-516C, Airworthiness Certification Criteria, Department of Defense Handboo, December 2014

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD-MM-YYYY)*
01-04-2020

2. REPORT TYPE
Contractor Report

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
Run Time Assurance as an Alternate Concept to Contemporary Development Assurance Processes

5a. CONTRACT NUMBER

5b. GRANT NUMBER
NNL16AA12B

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Peteron, Eric M.; Devore, Michael; Cooper, Jared; Carr, Greg

5d. PROJECT NUMBER

5e. TASK NUMBER
0LARC18F0193

5f. WORK UNIT NUMBER
340428.02.20.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/CR-2020-220586

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-
Subject Category 03
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES
Langley Technical Monitor: Wilfredo Torres-Pomales

14. ABSTRACT

NASA and the FAA sought industry research to identify and evaluate alternate concepts for assuring safety of airborne systems. This report documents a research effort focused on the evaluation of Run Time Assurance (RTA) as applied to a novel, airborne system architecture. The RTA pattern is applied to a case study focused on a notional integrated flight and propulsion control system for a DEP VTOL aircraft. During flight, while the high-automation algorithms are operating, the RTA system will monitor the aircraft state for any impending violation of safety requirements. When necessary, it will switch to the low-automation software to prevent such violations. Assurance practices for both baseline industry activities and the RTA approach were captured and compared to illustrate the required engineering design considerations, and possible advantages and disadvantages of each approach as part of this case study.

15. SUBJECT TERMS
Assurance; Safety; VTOL

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	226	(757) 864-9658