Design and Testing of an Approach to Automated In-Flight Safety Risk Management for sUAS Operations

Ersin Ancel*, Steven D. Young[†], Cuong C. Quach[‡], Rafia F. Haq [§], Kaveh Darafsheh[¶], Kyle M. Smalling[‖], Sixto L. Vazquez**, and Evan T. Dill [††]
NASA Langley Research Center, Hampton, VA, 23681, USA

Ryan C. Condotta[‡‡] and Bailey E. Ethridge[§§]
Analytical Mechanics Associates, Inc., Hampton, VA, 23666, USA

Logan R. Teska[¶¶]
Kellogg, Brown & Root, Inc., Houston, TX, 77002, USA

Thomas A. Johnson***
Aerospace Innovations, LLC., Newport News, VA, 23606, USA

An onboard risk management automation design is presented based on run-time assurance principles, as well as the concept for In-Time Aviation Safety Management Systems (IASMS) as described by the National Academies. The automation is designed to operate independently of the autopilot and perform real-time risk assessment spanning multiple classes of hazards, predict constraint violations, and track autopilot states. In the event of elevated risk conditions or predicted constraint violations, the automation will select from a set of available contingencies and trigger autopilot mode changes if necessary to mitigate risk exposure. The onboard automation also informs the remote operator/pilot of what the independent monitor is observing and any contingency decisions or actions that may arise during flight. Details of an implementation of this design and results of verification and validation activities, as required to meet stringent NASA software and system assurance standards, are also presented. This includes simulation and flight testing using small unmanned aircraft systems.

I. Introduction

NASA's System-Wide Safety (SWS) project seeks to develop highly assured methods or approaches to proactively mitigate safety risks to future aviation operations. The primary motivation for the ongoing research is to develop means by which more timely actions may be taken to mitigate safety risk during operations, in alignment with the vision described in NASA's Aeronautics Research Mission Directorate's (ARMD) Strategic Implementation Plan [1] and the concept of In-Time Aviation Safety Management Systems (IASMS) as described by the National Academies [2]. In both visions, system safety awareness and provision are expanded through increased access to relevant data, integrated analysis and predictive capabilities, improved real-time detection and alerting of domain-specific hazards, decision support, and in some cases, automated risk mitigation strategies. Within SWS, such means are described as a collection of Services, Functions, and Capabilities (SFCs) that are supported by an underlying information system [3–7]. For example, an integrated risk assessment capability is envisioned that utilizes multiple functions and/or services to

*Aerospace Engineer, Aeronautics Systems Analysis Branch, MS 442, Member.
[†] Aerospace Research Engineer, Safety Critical Avionics System Branch, MS 234, AIAA Fellow.
[‡] Computer Engineer, Safety Critical Avionics Systems Branch, MS 234.
[§] Computer Engineer, Flight Software Systems Branch, MS 064.
[¶] Research Engineer, Safety Critical Avionics Systems Branch, MS 234.
[‖] Aerospace Engineer, Safety Critical Avionics Systems Branch, MS 234.
**Electrical Engineer, Safety Critical Avionics Systems Branch, MS 234.
[††] Aerospace Research Engineer, Safety Critical Avionics Branch, MS 234, Member.
[‡‡] Software Developer, Aeronautics Systems Analysis Branch, MS 442.
[§§] Software Developer, Aeronautics Systems Analysis Branch, MS 442.
[¶¶] Applications Developer, Flight Software Systems Branch, MS 064.
***Chief Engineer, Flight Software Systems Branch, MS 064.

monitor safety-related metrics and margins and recommend timely operational changes. Assessments may be based on data analytics and predictive models derived from heterogeneous data sets, spanning a number of relevant indicators and their time histories.

Initially, a concept of operations was developed as well as preliminary architecture and information requirements for two emerging application domains [4, 7]. These domains comprise highly autonomous operations at low altitudes, over populated areas, and using (1) small unmanned aircraft systems (sUAS) or (2) larger vertical takeoff and landing (VTOL) aircraft suitable for air taxi operations. Both domains are characterized as continuous autopilot-based flights with supervisory oversight from a remote location and mission times no longer than 20 minutes. Focusing on these domains allows for proof-of-concept demonstration and exposure of requirements for SFCs intended to have significant benefit in terms of safety risk mitigation and cost-effectiveness. An integrated UAS traffic management (UTM)-based service coordinates airspace access and supports separation assurance.

It should be noted that some of the SFCs being investigated are at a low Technology Readiness Level (TRL), with advancing TRL as one goal of the ongoing research. Also, in terms of architectural requirements [4, 7], an important trait for safety-critical systems (such as described later in Section II) is to provide some level of Run-Time Assurance (RTA) capability [8–11]. RTA approaches are based on (1) independent monitoring and (2) a monitor design that is less complex and at a higher assurance level (i.e., less likely to fail) than the system it is monitoring (in this case a commercial-off-the-shelf (COTS) autopilot). For sUAS, this approach is part of an industry standard for bounding the behavior of complex unmanned aircraft [12]. RTA techniques are also referred to as run-time verification (RTV), where the focus is on software behavior verification. One of the applied RTA methods monitors safety properties, which are abstracted and desired safety characteristics of the vehicle's operation [9].

Across the SWS portfolio, more than 20 IASMS SFCs are being investigated, with more planned for the future tailored to domain-specific safety risks that have not been well-addressed by SFCs in existing commercial products. This paper describes the subset of services and functions that have been developed and integrated to evaluate a capability for automated risk mitigation by an onboard system. The remainder of this paper will describe the initial design and implementation (Section II), verification and validation activities, including simulation and flight testing (Section III), and concluding remarks and next steps (Section IV). For additional SWS background and project-specific definitions of terms, see Refs. [3, 4, 6]. Companion papers on related work can be found in [7, 13–16].

II. Automated In-flight Safety Risk Management Architecture Design and Implementation

Several in-flight risk management approaches have been investigated in recent years. Similar to the approach described in this paper, Ref. [17] proposed a decision-making framework that takes into account combined risks of battery power loss and obstacle collision to assess the probability of mission failure, then replans the vehicle trajectory if failure risk exceeds a predetermined threshold. In a parallel effort, Ref. [18] developed a finite state machine approach within a modular onboard architecture which monitors vehicle health parameters, such as navigation and communication subsystems, and evaluates path feasibility to assess mission risk and trigger a mitigation action corresponding to the current flight state. The Assured Contingency Landing Management capability presented by Ref. [19] introduces a sophisticated contingency management approach which analyzes vehicle controllability and landing site reachability to select a landing strategy which minimizes cost as a function of population, airspace density, and landing site proximity. Finally, Ref. [20] applied an extension of Dempster-Shafer theory, using auto-updating joint conditional probability matrices, to translate individual subsystem health statuses to a probabilistic estimate of vehicle-level risk. The evidential combination approach was compared against more traditional contingency management approaches and found to be more effective in detecting intermittent failures. In contrast, the approach described in this paper is designed to align with the IASMS concept, consider risk across a user-defined set of metrics, provide an increased level of assurance, and bound behavior when executing contingencies.

The SFCs comprising an initial implementation of an automated in-flight safety risk management capability such as envisioned in Section I, are distributed across both ground and aircraft systems, as shown in Fig.1. Onboard functions are implemented as a flexible set of applications running on the core Flight System (cFS) framework [4, 21]. The applications utilize the cFS software bus (SB) to exchange messages with each other and with external systems or functions. A set of bus bridge applications (e.g., Safeguard Bridge, CANBus Bridge, and MavLink Bridge) translate serial data streams to and from external systems. This includes, for example, sensors and the COTS autopilot. A description of each function shown in Fig.1 is provided in the following sections.

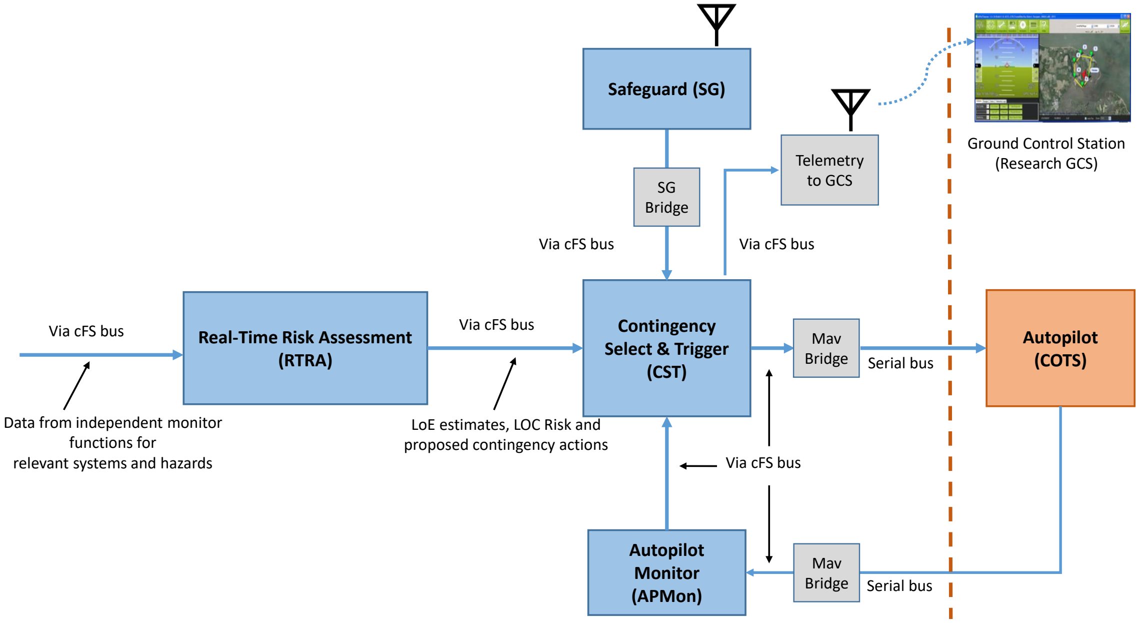
Safeguard (SG)
SG Bridge
Via cFS bus
Telemetry to GCS
Via cFS bus
Ground Control Station (Research GCS)
Via cFS bus
Real-Time Risk Assessment (RTRA)
Via cFS bus
Contingency Select & Trigger (CST)
Mav Bridge
Serial bus
Autopilot (COTS)
Data from independent monitor functions for relevant systems and hazards
LoE estimates, LOC Risk and proposed contingency actions
Via cFS bus
Autopilot Monitor (APMon)
Mav Bridge
Serial bus

Fig. 1 Onboard Risk and Contingency Management Automation Architecture.

A. Real-Time Risk Assessment (RTRA)

The RTRA function presented in this paper is an extension of the previously developed in-time Non-participant Casualty Risk Assessment (NPCRA) framework documented in Refs. [21] and [22]. NPCRA originally consisted of three separate modules that utilized real-time aircraft health and environmental data to estimate the risk to populated areas on the ground due to flight-critical failure onboard the aircraft. These modules include 1) a probabilistic graphical model that outputs mishap likelihood, 2) an off-nominal trajectory and impact point prediction model that estimates the trajectory and crash location following a failure, and 3) a severity estimation model that uses a combination of impact point location, high-resolution dynamic population density data, roof penetration models, and other onboard databases to determine the probability of one or more human casualties[*] [21]. As part of the iterative development process, the NPCRA framework was extended to include an updated severity estimation module with augmented population density information that employs both cellular tower data and badge reader data from NASA Langley Research Center. Additionally, the likelihood model was expanded to incorporate several aircraft sensors and onboard SFCs to estimate additional Likelihood of Events (LoEs) estimates beyond the loss of control (LOC) risk estimate. Finally, the RTRA code was modified to output individual LoEs as well as recommended mitigation actions to the cFS SB.

1. RTRA Likelihood Model

The likelihood component of the RTRA framework is modeled using a Bayesian Belief Network (BBN) approach and implemented in HUGIN Expert software [24]. BBNs are directed, acyclic graphs which are widely used to conduct probabilistic reasoning in uncertain, complex domains. Graph nodes represent random variables and directed edges denote the dependent relationships between them. Variable relationships are defined by conditional probability tables (CPTs), which can be specified using empirical data, subject matter expert (SME) opinion, or a combination of the two [25]. For background information on the chosen modeling framework and software, see Ref. [22].

The RTRA BBN was designed to represent an octocopter research aircraft (see Ref. [7] for more details on the test vehicle and platform), where several health parameters are ingested at a rate of approximately 1 Hz to calculate real-time status estimates of various vehicle subsystems. Five submodels compare these parameters against softcoded thresholds, read from a mission configuration file at time of flight, in order to estimate probabilities of adverse outcome events (or LoEs): loss of navigation, loss of command & control (C2) link, loss of Mission Planner (MP) link, research hardware power loss, and LOC (Fig. 2). These probabilities are then used to propose mitigation action upon anomaly detection.

[*] Within this research, a casualty is considered to be a serious injury or fatality [23].

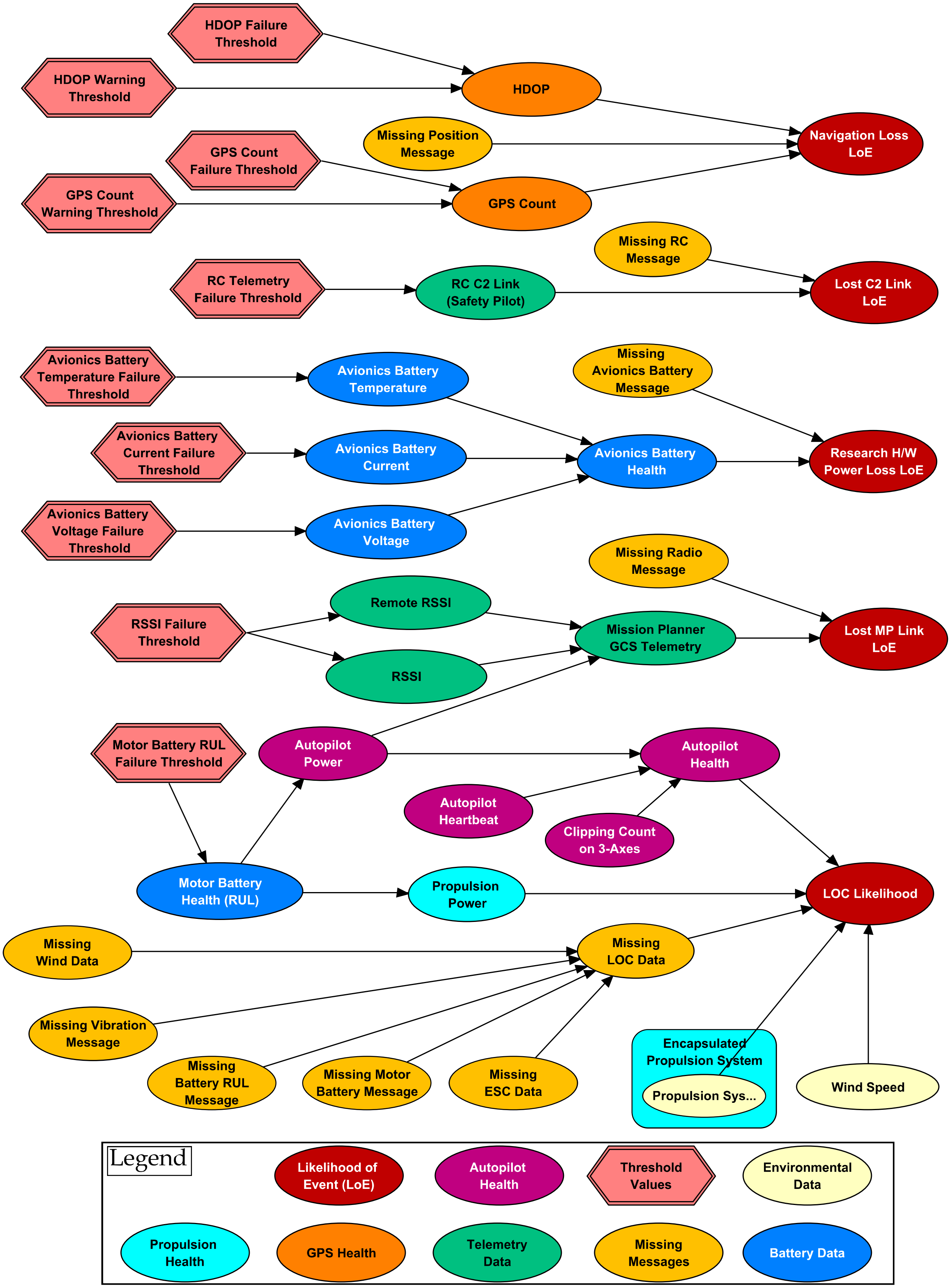

Fig. 2 RTRA Bayesian Belief Network.

The navigation subsystem reads in a GPS satellite count and a horizontal dilution of precision (HDOP) value. The GPS satellite count and HDOP nodes are assigned one of three category states (poor, fair, good) by checking the values against warning and failure thresholds. In-house SME opinion is used to assign a probability of navigation loss to each of its two parent nodes in the manually specified CPT. The C2 link subsystem receives one input value, radio control (RC) telemetry, which is compared against a softcoded failure threshold. A RC telemetry value falling below this threshold indicates a lost RC link and consequently a loss of C2 link. The MP link subsystem has two input nodes: Received Signal Strength Indication (RSSI) and remote RSSI. RSSI and remote RSSI are both checked against the same softcoded failure threshold then assigned a Boolean state (failure, nominal). Any one of these two parent nodes having a failure state indicates a lost Mission Planner link. The research hardware power subsystem receives avionics battery health information as inputs. Three parameters—battery voltage, current, and temperature—are checked against corresponding threshold values, then each node is assigned either a failure or nominal state. A failure state for any of these inputs signifies an avionics battery failure, and consequently, research hardware power loss.

The final submodel, loss of control (LOC), is informed by the ground control station (GCS)-provided dynamic wind speed data in addition to health information received from three vehicle subsystems: autopilot health, propulsion power, and propulsion system health. The autopilot health node assumes either a failure or nominal state, determined by clipping count[†] and the autopilot heartbeat frequency. The propulsion power node also assumes either a failure or nominal state based on the state of its singular parent node, motor battery health. The motor battery health information is obtained from the onboard battery health diagnostics that provide real-time Remaining Useful Life (RUL) data from a separate battery prognostic service [26]. The propulsion system node (represented as a separate, encapsulated subnet, given in Fig. 3) has five parent groups: electronic speed controller (ESC) voltage, ESC current, ESC temperature, ESC motor revolutions per minute (RPM), and vibration. Each of these nodes assumes either a failure, warning, or nominal state, then the propulsion system status node's state is determined by an expression that prioritizes the most severe state on any parent subsystem. Like the navigation subsystem, the LOC CPT was manually populated by project SMEs. Finally, each LoE construct described above receives input from a missing message node that provides the refresh frequency status of the respective cFS messages, checked against a time constant. If the messages are not updated within the specified time frame, the associated LoE will indicate that the system is no longer capable of providing a valid mitigation action by sending a missing data message.

In addition to the threats discussed above, another onboard application, Proximity to Threat (PtT), provides awareness of known static (buildings, poles, trees, or terrain) and/or dynamic (temporary flight restriction, adverse weather) threats [13, 27, 28]. The application publishes a list of violations to the software bus if the flight path approaches such threats closer than a predefined threshold value. Each published violation includes the threat type, distance to threat, and the Safety Margin (SM), expressed as a percentage of the threshold value. Specifically, the Safety Margin is a function of the threshold distance, where a SM of 0% indicates that the aircraft is at the threshold distance from the obstacle, a SM of 100% indicates that the distance is twice the threshold value, and a SM of -100% indicates that a collision between the obstacle and the aircraft is imminent. By subscribing to the PtT messages, the RTRA application can interpret the violations and issue mitigation actions when the SM $\leq$ 0%. The corresponding mitigation actions for RTRA-derived LoEs, as well as likelihood of proximity risk, are provided in Section II.A.2.

[†] Clipping count value indicates the number of times the vibration amplitude exceeds 60 m/s^2, which leads to accelerometer saturation.

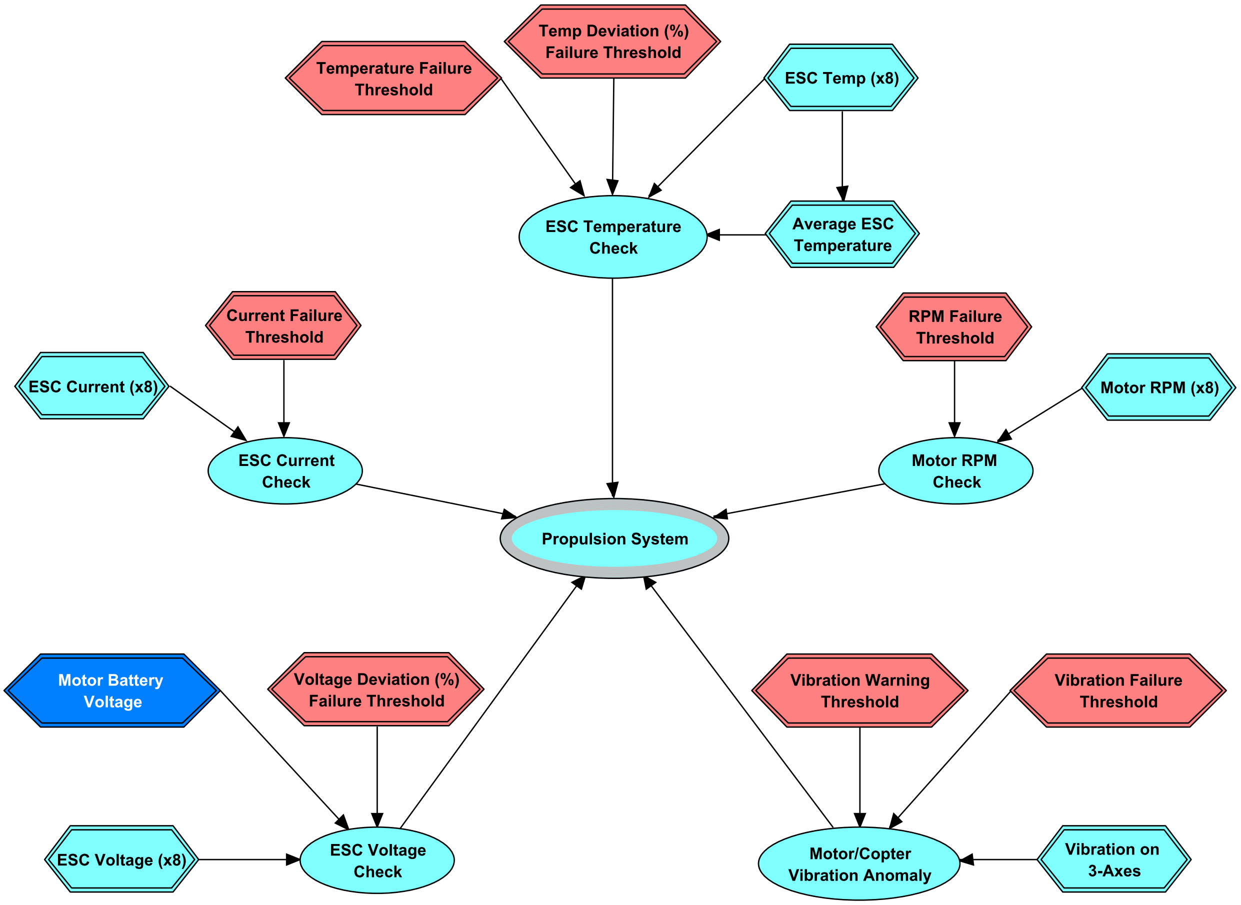

Fig. 3 Encapsulated Propulsion System Bayesian Belief Network.

2. RTRA Mitigation Actions

The RTRA function recommends an aggregate mitigation action based on various LoE probability values. The LOC mitigation action is determined via a lookup table that classifies severity (columns) and likelihood (rows) values to provide a categorical estimate of risk. The severity of the mishap, measured by the probability of non-participant casualty (P_C), is estimated by a separate module within RTRA. The resulting lookup table (or risk matrix), given in Table 1, then specifies the appropriate mitigation action. The mitigation actions for loss of navigation, loss of C2 link, loss of Mission Planner (MP) link, research hardware power loss, and proximity risk are provided in Table 2.

Table 1 Loss of Control Risk Representation and Respective Mitigation Actions.

Severity/P_C ➡ LOC Likelihood ⬇	Minimal $0 \leq P_C < 0.25$	Minor $0.25 \leq P_C < 0.50$	Major $0.50 \leq P_C < 0.75$	Catastrophic $0.75 \leq P_C \leq 1$
Frequent $0.5 \leq P_{LOC} \leq 1$	Land	Land	Land	Land
Probable $0.1 \leq P_{LOC} < 0.5$	NOP	Land @ Vertiport/RTL	Land @ Vertiport/RTL	Land
Remote $0.01 \leq P_{LOC} < 0.1$	NOP	Land @ Vertiport/RTL	Land @ Vertiport/RTL	Land @ Vertiport/RTL
Improbable $0 \leq P_{LOC} < 0.01$	NOP	NOP	NOP	Refer to GCS Operator/NOP

Table 2 Mitigation Actions for Various LOEs.

Likelihood of Event (LoE)	Proposed Mitigation (Priority)
Likelihood of Navigation Loss	Land (1)
Likelihood of Lost C2 Link	Land @ Vertiport/RTL (2)
Likelihood of Lost MP Link	Position Hold (3)
Likelihood of Mission/Data Loss	Land @ Vertiport/RTL (2)
Likelihood of Proximity Risk	Position Hold (3)

RTRA considers the five LoE probabilities to recommend a single mitigation that is proposed via the cFS bus to the Contingency Select & Trigger (CST) application (to be discussed in Section II.D). If multiple LoEs exceed their threshold simultaneously, the mitigation is selected based on a prioritization order defined a priori by the operator/user. The order used for testing is given in Table 2, with (1) being the highest priority where the commanded action is to land the aircraft immediately. The second highest priority action is 'Land at Vertiport/Return to Launch' (RTL), followed by 'Position Hold', then 'No Operation' (NOP, 'Do Nothing', or 'Continue'). In this context, the NOP action signifies no change to the current path. Using the prioritization order, RTRA selects the highest priority mitigation action and publishes it to the cFS SB.

B. Safeguard Capability

1. Safeguard Function

Safeguard is an independent onboard system that warns of impending violations of geo-spatial constraints (e.g., Stay-in and Stay-out areas) as well as other constraints (e.g., range limits, path deviations, and altitude/airspeed restrictions). The system operates on isolated hardware and is qualified at the NASA Class B Safety-Critical level [4, 29, 30]. This is the assurance level required of safety-critical software on unmanned NASA spacecraft or large scale aeronautics vehicles [31]. In the context of testing the risk mitigation system described here, Safeguard's main objective is to ensure that the vehicle remains within the defined flight area. This is accomplished by defining a geo-fence polygon which, when approached by the vehicle, results in warning signals generated by Safeguard. The system operates at 5 Hz and asynchronously from the other elements shown in Fig. 1.

During flight, Safeguard continuously computes the predicted vehicle impact trajectory in the event of total power loss. If the predicted trajectory crosses defined boundaries (i.e., geo-fence polygon edges), two types of flags, Warning and Critical Warning, can be issued. The Warning flag allows the user to adjust a scale factor to increase warning time/distance to a defined boundary. In contrast, a Critical Warning flag[‡] is used to provide assurance that a defined boundary will not be breached upon flight termination. For the application described in this paper (as shown in Fig. 1), Safeguard flags are then used by the CST function to decide the appropriate contingency (i.e., autopilot mode change).

2. Safeguard Bridge (SGBridge) Application

SGBridge is a cFS application executing on a 10 Hz schedule to receive serial data packets from an external Safeguard unit. The SGBridge receives Safeguard output packets, verifies their validity, and publishes all valid packets to the SB for other cFS applications to utilize. The external Safeguard unit produces a single (fixed binary structure) output packet at 5 Hz in operational mode (corresponding to each solution) and at 1 Hz in configuration mode (providing navigation data without a Safeguard solution). To support the testing environment, the SGBridge application processes ArduPilot playback messages that allow specific parameters in the Safeguard packet to be overwritten. This feature is only available when the application is executed in the simulation environment. The SGBridge application outputs 38 unique Safeguard-generated values based on various vehicle parameters including position, altitude, health, and bank angle. Like the RTRA mitigation actions (Land, Land at Vertiport/RTL, Position Hold, and NOP), each Safeguard-generated warning has a corresponding mitigation action that is passed to the CST function.

[‡] The default action for a Critical Warning flag is flight termination, however, within the context of this flight campaign, the recommended mitigation action is Land.

C. Autopilot Monitor (APMon) Function

For the current design, Autopilot Monitor (APMon) function is intended to help assure that the mitigating action recommended by the RTRA/Safeguard/CST functions meets safety properties associated with the autopilot design and operation. In other words, the system should not command the autopilot to change to an unsafe or unavailable mode given its current state. This run-time monitor application, which also executes in the cFS framework, is derived from code generated by and based upon safety properties expressed in the Co-Pilot formalism [8]. Co-Pilot is an emergent formal methods technology increasingly used in safety-critical system design to assure deterministic and safe behavior of complex systems.

D. Contingency Select and Trigger (CST) Function

CST is a decision-making application that processes data received from other applications and uses deterministic logic conditions to inform the autopilot of the desired mode change, if any, to execute (e.g., Land). This mode change, or mitigation action, is then published to the cFS SB for use by the autopilot. CST receives and processes information from three applications and bridges: RTRA, SGBridge, and APMon. As previously discussed, RTRA provides CST with data about the likelihood and severity of an undesirable event occurring, as well as any recommended mitigation actions. SGBridge provides CST with Safeguard Warning and Critical Warning flags through a serial interface. Once CST receives and processes data from RTRA and SGBridge, it is processed by a series of logic conditions to determine the appropriate action based on the vehicle's environment and the prioritization order given in Section II.A.2. Before the mitigation action can be sent to the autopilot, CST needs to ensure that the proposed solution is available. To do so, the CST application checks the list of available autopilot modes provided by the APMon application. Once the proposed mitigation's availability is confirmed, the aggregate mitigation action is relayed to the autopilot for execution.

E. Telemetry and Ground Control Station (GCS) Functions

These functions allow for operators, pilots, and/or researchers to observe the state of the onboard elements. They also allow on-board measurements and alerts to be propagated to ground-based SFCs, thereby enabling displays which combine aircraft status with ground information context. Within the SWS research portfolio, customized prototype GCS software were developed and tested to support human factors-centric research, as discussed in companion papers, Refs. [14] and [15].

III. Testing

The onboard functions presented in this paper underwent rigorous testing in accordance with NASA software and system development standards, specifically the NASA Procedural Requirements (NPR) 7150.2D standards for the appropriate software classifications[§] [31]. In addition to fulfilling requirements to gain flight test approval, these testing procedures enabled realization of an RTA approach like that described in Refs. [8–11, 16]. As outlined in Ref. [12] and Ref. [16], the goal of the applied approach is to safely bound the behavior of a system that is at an unknown assurance level with one that is tested to a high assurance level. Testing was performed in both simulated and flight environments using multiple test capabilities to achieve the necessary rigor: 1) a software-in-the-loop (SITL) playback capability that used the same flight computers, with sensor information obtained from log files (Section III.A.1), or modified to test specific functionality (Section III.A.2), and 2) dual identical computers onboard (Section III.B); one isolated from the autopilot for initial testing and analysis of logged outputs, and the other for final testing as connected to the autopilot. Results of preliminary flight testing are given in Section III.B. A complete set of results will be published in the future after all tests are finalized.

A. Software-in-the-loop (SITL) Testing

1. Playback Capability

A playback capability was created to enable unit and integration testing of the onboard software. The playback simulation is a way to test the behavior of the entire code chain, beginning at the cFS code on the aircraft all the way to the SWS Research Ground Station and the Supplemental Data Service Providers (SDSPs). The playback simulation conceptually plays back recorded data from previous flights onto the software bus in the cFS environment. This enables

[§] RTRA and CST software are being developed to Class C Non-Safety Critical.

software applications under testing to 1) read their required input from the software bus, 2) perform their computations, and 3) output their messages, which get downlinked to the SWS Research Ground Station and onward to the SDSPs. The playback capability is comprised of three applications which provide a synchronized raw data stream on the cFS SB: 1) Controller Area Network (CAN) Bridge Playback, 2) Ardupilot Playback, and 3) Playback Manager.

The first two apps were designed to replay in-flight data captured from the CAN Bridge Battery and Ardupilot systems, respectively. The third app was designed to manage and synchronize the timing of playback application operation and communication to other cFS applications, particularly the Ground Communication application (GCOM), which is used for relaying playback data to the SWS Research Ground Station. Both CAN Bridge Playback and Ardupilot Playback act as direct substitutions for their counterpart CAN Bridge and Ardupilot applications that capture in-flight data from hardware systems. By modifying previously recorded flight data via the Flight Log Content Editor (FLCE) (described in the following section), the playback tool allows for simulating various off-nominal scenarios to replicate mitigation action responses of the entire risk management architecture.

2. Flight Log Content Editor (FLCE)

The FLCE tool was developed to generate and manipulate flight data logs to be used with the playback capability. The application has the ability to modify several Mavlink messages (heartbeat, aircraft position, navigation health, telemetry health, Mission Planner health, vibration, clipping, and ESC voltage, current, RPM, and temperature values), as well as 38 unique Safeguard warning and fault messages. FLCE allows users to select which variable they want to modify, new variable value, and how long after takeoff they want the change to occur using the graphic user interface.

3. RTRA Testing

A total of 92 possible BBN model configurations were bench tested using the playback simulation capability. This set of test cases was chosen to be representative of all possible outcomes and thoroughly test all CPTs and expressions without redundancy. Given the independence of the five subsystems in the test build model, it was not necessary to test combinations of state configurations outside of each individual subsystem. For the navigation, C2 link, and Mission Planner link subsystems, simulating the complete list of state configurations was feasible due to the comparatively low number of nodes. Due to the sizeable number of possible combinations for LOC parent node states, a testable subset of this subsystem's configurations was derived using a bottom-up approach. First, an exhaustive list of configurations for each root-level parent group was generated and then down-selected to meet minimum testing criteria and fulfill the identified subset of their children nodes' possible configurations. This process was repeated until the leaf (LoE) node was reached, thereby testing each expression/CPT case and each node state configuration without expending the computational resources required to test an exhaustive list. The resulting 92 cases were simulated by running modified flight logs, generated by the FLCE software, using the playback feature. Simulation output was checked against expected results, with the first pass exposing a design error. Once resolved, all cases were retested successfully with the results independently verified.

4. CST Testing

Like the RTRA bench tests described above, the CST software also underwent unit testing to ensure proper operation with the software it interacts with, namely SGBridge, RTRA, APMon, and, indirectly, the autopilot. The RTRA-CST interaction was tested by first identifying a subset of RTRA test cases to simulate potential RTRA mitigation actions. These cases were then used as the input conditions for CST using the playback feature, and simulation outputs were verified against expected results. Additionally, RTRA also provides its own mitigation action recommendations as well as raw data for CST to draw its own conclusions. Using the raw data, CST is able to cross-check the recommended mitigation action received from RTRA with the mitigation action recommended by CST itself. If both applications arrive at the same command, then CST publishes the command to cFS after checking APMon. If the recommended command from RTRA is different than CST's, then CST's command takes precedence and is published to cFS after checking APMon.

The CST-SGBridge interaction was also tested using the playback feature. This was done by artificially simulating primary failures and warnings from seven main Safeguard Warning categories and by comparing the CST mitigation actions against the expected results. Finally, the CST-APMon interaction was simulated. CST receives a list of available modes from APMon. After evaluating RTRA and SGBridge outputs and determining a mitigation action, CST checks APMon's list to ensure that the suggested mitigation is indeed available before publishing the action to the cFS SB.

B. Flight Testing and Results

1. Simulated Flight Test Scenario

To assess technology maturity, test scenarios were defined spanning four use cases and two test ranges or locations as described in companion paper [7]. For brevity, only the vertiport approach/departure use case testing is discussed here. To test several SFCs simultaneously, a combined test matrix comprised of various simulated failures, warnings, and violations were implemented and tested in a virtual city. The flight test scenario presented in Fig. 4 involves a vertiport approach/departure flight path over a densely populated urban setting with numerous virtual and real structures near the route of flight. The following section provides results for a representative flight test that was completed on March 16, 2022. During this test, safety metrics thresholds were adjusted so as to trigger various LoE conditions; these include: LOC and its underlying causes, nonparticipant casualty, proximity to threat, and Safeguard violation.[¶] The labeled points (A-G) along the flight path in Fig. 4 indicate where these events or conditions occurred. Note that this test was not conducted with commands sent to the autopilot. Planned tests will exercise sending mode change commands to the autopilot to demonstrate mitigating actions. In these upcoming tests, the mitigation action will be performed once selected thus necessitating a reset after each simulated failure.

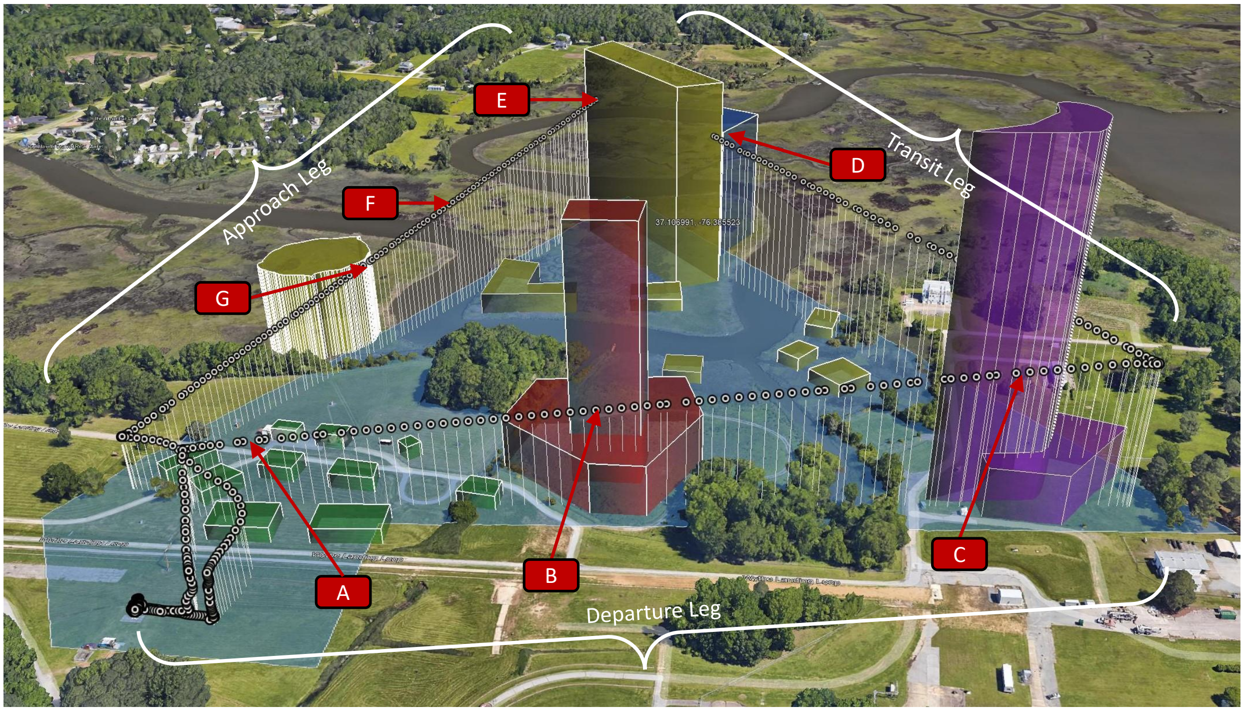

Fig. 4 Vertiport Approach/Departure Scenario Flight Path over Simulated Urban Area.

2. Flight Test Results

The flight path selected for this scenario (Fig. 4) was a triangular path which represents the departure and arrival legs of a longer air taxi operation, as well as a transition leg from the departure to the arrival. The flight begins in the southwestern corner of the triangle and proceeds in a counter-clockwise pattern totaling approximately 1.6 km and reaching an altitude of 104 m above ground level at the northernmost waypoint. The outbound segment was planned to match operations representative of those in an urban environment which is envisioned to require a near-vertical climb until nearby buildings and treetops are cleared, at which time a gradual climb to the desired operating altitude is achieved. A reversed altitude profile was used for the inbound segment. Tests are designed such that LoE exceedences should occur at various locations, denoted A-G, during the flight; results are given in Table 3 and data corresponding to the sampled locations are given in Fig. 5.

[¶] Previously mentioned LoEs such as navigation, C2, Mission Planner (or telemetry), and research hardware power loss were simulated using the playback feature and were not demonstrated in flight.

Table 3 Risk Values and Associated Mitigation Actions for Sample Locations.

Sample Location	Lat/Long Altitude	LOC Likelihood	Probability of Casualty (P_C)	Population Density (person/100m^2)	Propulsion System Risk	Proximity Risk (Safety Margin)	Safeguard Violation	Recommended Mitigation Action
A	37.1031, -76.3873, 34.24 m	0.10	0.31	5	0.00	100.00%	NO	RTL
B	37.1033 , -76.3852, 54.00 m	0.10	0.00	10	0.00	10.06%	NO	NOP
C	37.1036, -76.3829, 75.78 m	0.10	0.00	10	0.00	-69.65%	NO	POSHOLD
D	37.1063, -76.3848, 98.65 m	1.00	0.71	20	0.00	-23.12% 90.12%	NO	LAND & POSHOLD → LAND
E	37.1072, -76.3860, 96.03 m	1.00	0.83	30	0.00	-56.66%	YES	LAND & POSHOLD & LAND → LAND
F	37.1055, -76.3869, 69.58 m	1.00	0.84	30	1.00	100.00%	NO	LAND & NOP → LAND
G	37.1048, -76.3872, 58.32 m	1.00	0.27	30	0.00	-26.40%	NO	LAND & POSHOLD → LAND

- Location A: As previously described, the mitigation action for exceeding a user-defined LOC risk threshold is determined by the look-up table provided in Table 1 within Section II.A.2. The BBN-derived LOC likelihood value of 0.1 is classified as 'Probable' due to simulated high winds (above 5 m/s), and the estimated P_C value of 0.31 is categorized as 'Minor' considering the unprotected population density beneath the flight path. The LOC risk look-up matrix depicts the mitigation action at location A to be Land @ Vertiport/RTL as all other simulated parameters (propulsion system, proximity risk, and Safeguard violation) are nominal.
- Location B: Although the LOC likelihood remains the same as the previous location, the P_C value was computed as 0 due to sheltering effect provided by the building located at the aircraft impact point (not shown). Per Table 1, the proposed mitigation action is NOP (Continue; no change to autopilot mode). Additionally, the PtT application issues a violation for the nearby building, however, as the SM value (10.06%) remains above 0%, RTRA/CST does not recommend a Position Hold mitigation action.[‖]
- Location C: The PtT application triggers a proximity violation as the vehicle crosses the threshold boundary while approaching the building from its southwest corner. The proposed mitigation for proximity violation is Position Hold as the SM value falls below 0% (-69.65%) (see Table 2). CST issues Position Hold given all other threats register below actionable limits. Note that the P_C remains 0 due to sheltering effects of the building.
- Location D: Another PtT test involves navigation between two buildings (i.e., urban canyon) where the application issues two proximity violations triggered by each building (-23.12% and 90.12%). Given that the SM falls below 0% for at least one of the buildings, the Position Hold mitigation action is issued. Simultaneously, the LOC likelihood reaches 1.0 due to motor battery RUL[**] falling below the 120 second threshold (Fig. 5). Combined with the P_C value (0.71), the recommended LOC mitigation action is Land. Using the prioritization scheme, between the Land and Position Hold commands, CST issues Land mitigation to the autopilot.
- Location E: At this location, the LOC likelihood remains at 1.0 due to RUL continuing to decrease below the set threshold. Additionally, the high population density below the flight path results in a P_C of 0.83; both values trigger Land mitigation actions. Simultaneously, the PtT application issues a Position Hold due to a proximity violation with a -56.66% SM. Finally, the Safeguard system issues two flags, first a Warning flag, indicating that the vehicle is approaching the Stay-in boundary (shown in light blue in Fig. 4, set deliberately close to the flight path), followed by a Critical Warning flag once the boundary is crossed. The Safeguard Warning and Critical Warning flags prompt CST to issue Position Hold and Land, respectively. Of the actions recommended by the three input systems, Land is prioritized and sent by CST as a command to the autopilot.

[‖] For this test, the PtT threshold value was set to 50 ft, consequently a 0% SM indicates the aircraft is 50 ft from the obstacle.

[**] At the time of writing, the RUL parameter required further calibration to represent a fully charged battery and to eliminate noise.

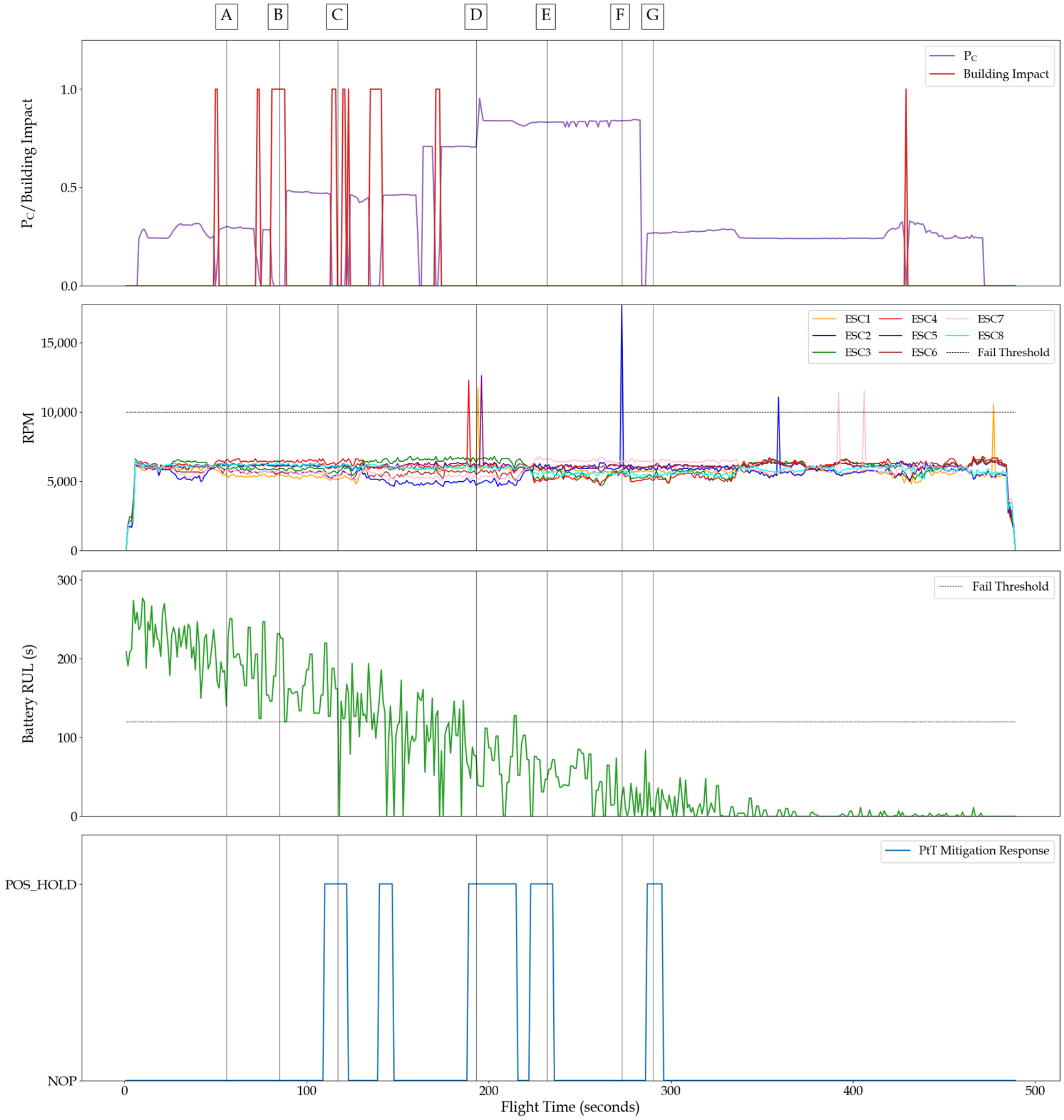

Fig. 5 ***P_C*, Motor RPMs, RUL, and PtT Mitigation Response Corresponding with Sample Locations.**

- Location F: For the remainder of the flight, the RUL fault-derived high LOC likelihood continues to trigger the Land command. Additionally, at this location, the propulsion system issues a transient warning due to a single motor exceeding the RPM warning threshold (ESC2).[††] Between the NOP (originating from propulsion warning) and Land (caused by RUL violation), the recommended action at location F remains Land.
- Location G: The final location involves a PtT application response against the side of a building. Although the PtT application starts tracking the building earlier in the trajectory, the proximity risk mitigation action (Position Hold) is only issued once negative SM values are published (e.g., -26.40%). Similar to location F, the Land mitigation is recommended due to prioritization order.

[††] Per octocopter design, an anomaly on a single motor/ESC triggers a propulsion system warning with no mitigation action (NOP), where anomalies observed on more than one motor/ESC trigger a propulsion system failure which corresponds to a Land mitigation response.

IV. Concluding Remarks and Next Steps

Future advanced safety management systems seek to include more timely risk mitigation capabilities. These "in-time" capabilities may perform on the order of a few days to a few seconds, and may be supervised or fully automated. In this paper, an automated in-flight safety risk mitigation design is presented based on this concept and the application of run-time safety assurance principles. While it is true that current sUAS autopilot systems are designed with some degree of auto-mitigation, these are condition-based, limited to a few known (or expected) failure conditions, and not certified to high assurance levels. For example, most sUAS autopilots will simply switch to an emergency RTL mode in the event of lost link. In contrast, the system presented here is subjected to extensive V&V processes, operates independently of the autopilot, and is based on real-time risk assessment spanning multiple classes of hazards, prediction of constraint violations, and tracking of autopilot states. In the event of elevated risk conditions or predicted constraint violations, the independent system will automatically select from a set of available contingencies and trigger autopilot mode changes if necessary to mitigate risk exposure. The system also informs the remote operator/pilot of what the independent monitor is observing to aid in supervisory oversight and intervention. Preliminary results of verification, validation, and evaluation activities support stringent NASA processes and standards such as are used for safety-critical spacecraft applications. Additional simulation and flight testing is planned across additional use-cases and platforms (i.e., vehicles and architectures) to advance this concept for bounding the behavior of future highly-autonomous aircraft.

Acknowledgments

The work presented here was supported by the SWS project in NASA ARMD's Airspace Operations and Safety Program (AOSP). The authors would like to thank the NASA Langley UAS Operations Office (UASOO) and Range Safety Officers, Mark Fry, Jennifer Fowler, and Scott Sims. The authors would also like to acknowledge Andrew Moore for his invaluable expertise and feedback on BBN and CST designs, and editing the manuscript. Finally, special thanks to Leslie Johnson, Ed Hogge, Dominic Gutierrez, Andrew Turner, Nick Rymer, and Sloan Glover for their software assurance and flight support.

References

[1] National Aeronautics and Space Administration, "NASA Aeronautics Strategic Implementation Plan: 2019 Update," URL: https://www.nasa.gov/aeroresearch/strategy, 2019, [retrieved on 3 October 2021].

[2] National Academies of Sciences, Engineering, and Medicine, *In-Time Aviation Safety Management: Challenges and Research for an Evolving Aviation System*, National Academies Press, 2018. doi:10.17226/24962.

[3] Young, S. D., Quach, C., Goebel, K., and Nowinski, J., "In-Time Safety Assurance Systems for Emerging Autonomous Flight Operations," *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018.

[4] Young, S. D., Ancel, E., Moore, A. J., Dill, E. T., Quach, C. C., Foster, J. V., Darafsheh, K., Smalling, K. M., Vazquez, S. L., Evans, E. T., Okolo, W. A., Corbetta, M., Ossenfort, J. P., Watkins, J., Kulkarni, C. S., and Spirkovska, L., "Architecture and Information Requirements to Assess and Predict Flight Safety Risks During Highly Autonomous Urban Flight Operations," NASA TM-2020-220440, 2020.

[5] Ellis, K. K., Krois, P., Koelling, J. H., Prinzel, L. J., Davies, M. D., and Mah, R. W., "A Concept of Operations (ConOps) of an In-time Aviation Safety Management System (IASMS) for Advanced Air Mobility (AAM)," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021. doi:10.2514/6.2021-1978.

[6] Kirkman, D., Mooberry, J., Reeser, R., Yang, M., Gould, K., Koelling, J., Davies, M., Ellis, K., Prinzel, L., Krois, P., Mah, R., and Infeld, S. I., "Informing New Concepts for UAS and Autonomous System Safety Management using Disaster Management and First Responder Scenarios," *IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021.

[7] Young, S. D., Ancel, E., Dill, E. T., Moore, A. J., Quach, C. C., Smalling, K. M., and Ellis, K. K., "Flight Testing In-Time Safety Assurance Technologies for UAS Operations," *Aviation Forum 2022*, American Institute of Aeronautics and Astronautics, 2022.

[8] Clark, M. A., Koutsoukos, X. D., Porter, J., Kumar, R., Pappas, G. J., Sokolsky, O., Lee, I., and Pike, L., "A Study on Run Time Assurance for Complex Cyber Physical Systems," Air Force Research Lab, Report No. ADA585474, 2013.

[9] Goodloe, A., "Challenges in High-Assurance Runtime Verification," *International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2016)*, Corfu, Greece, 2016, pp. 446–460. doi:10.1007/978-3-319-47166-2_31.

[10] Pike, L., Goodloe, A., Morisset, R., and Niller, S., "Copilot: A Hard Real-Time Runtime Monitor," *Runtime Verification*, edited by H. Barringer, Y. Falcone, B. Finkbeiner, K. Havelund, I. Lee, G. Pace, G. Roşu, O. Sokolsky, and N. Tillmann, Springer, Berlin, Heidelberg, 2010, pp. 345–359.

[11] Schierman, J., Cooper, M. D., Richards, N., Gandhi, N., Horneman, K., Smolka, S., Stoller, S., and Clark, M., "Run Time Assurance for Complex Autonomy," *Safe and Secure Systems and Software Symposium (S5)*, Dayton, OH, 2015.

[12] ASTM International, "Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions, ASTM F3269-17," 2017.

[13] Banerjee, P., Corbetta, M., and Jarvis, K., "Probability of Obstacle Collision for UAVs in Presence of Wind," *Aviation Forum 2022*, American Institute of Aeronautics and Astronautics, 2022.

[14] Ho, N., Johnson, W., Wakeland, K., Panesar, K., and Ancel, E., "Preliminary Risk Visualization Concepts for NASA In-Time Aviation Safety Management System," *Aviation Forum 2022*, American Institute of Aeronautics and Astronautics, 2022.

[15] Feldman, J. M., Martin, L., Bradley, J. A., Walter, C. M., and Gujral, V., "Developing a Dashboard Interface to Display Assessment of Hazards and Risks to sUAS Flights," *Aviation Forum 2022*, American Institute of Aeronautics and Astronautics, 2022.

[16] Neogi, N. A., Young, S. D., and Dill, E. T., "Establishing the Assurance Efficacy of Automated Risk Mitigation Strategies," *Aviation Forum 2022*, American Institute of Aeronautics and Astronautics, 2022.

[17] Quiñones-Grueiro, M., Biswas, G., Ahmed, I., Darrah, T., and Kulkarni, C., "Online Decision Making and Path Planning Framework for Safe Operation of Unmanned Aerial Vehicles in Urban Scenarios," *International Journal of Prognostics and Health Management*, Vol. 12, No. 3, 2021. doi:10.36001/ijphm.2021.v12i3.2953.

[18] Baculi, J. E., and Ippolito, C. A., "Onboard Decision-Making for Nominal and Contingency sUAS Flight," *AIAA Scitech 2019 Forum*, 2019.

[19] Kim, J., Sharma, P., Atkins, E., Neogi, N., Dill, E. T., and Young, S. D., "Assured Contingency Landing Management for Advanced Air Mobility," *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–12. doi:10.1109/DASC52595.2021.9594498.

[20] Dunham, J., Johnson, E., Feron, E., and German, B., "Unmanned Systems Health Analysis through Evidential Reasoning Networks," *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, 2020, pp. 1–10. doi:10.1109/DASC50938.2020.9256593.

[21] Ancel, E., Capristan, F. M., Foster, J. V., and Condotta, R. C., "In-Time Non-Participant Casualty Risk Assessment to Support Onboard Decision Making for Autonomous Unmanned Aircraft," *AIAA Aviation 2019 Forum*, American Institute of Aeronautics and Astronautics, Dallas, TX, 2019. doi:10.2514/6.2019-3053.

[22] Ancel, E., Capristan, F., Foster, J. V., and Condotta, R., "Real-time Risk Assessment Framework for Unmanned Aircraft System (UAS) Traffic Management (UTM)," *Aviation Technology, Integration, and Operations (ATIO) Conference*, American Institute of Aeronautics and Astronautics, Denver, CO, 2017. doi:10.2514/6.2017-3273.

[23] Federal Aviation Administration, "Flight Safety Analysis Handbook (Version 1.0)," URL: https://www.faa.gov/about/office_org/headquarters_offices/ast/media/Flight_Safety_Analysis_Handbook_final_9_2011v1.pdf, 2011, [retrieved on 20 April 2022].

[24] Hugin Expert, "Hugin Developer, Version. 8.9," Aalborg, Denmark, 2020.

[25] Korb, K. B., and Nicholson, A. E., *Bayesian Artificial Intelligence*, 2nd ed., Taylor & Francis, 2010.

[26] Daigle, M., and Kulkarni, C. S., "Electrochemistry-Based Battery Modeling for Prognostics," *Annual Conference of the Prognostics and Health Management (PHM) Society 2013*, New Orleans, LO, 2013. doi:10.36001/phmconf.2013.v5i1.2252.

[27] Corbetta, M., Banerjee, P., Okolo, W., Gorospe, G., and Luchinsky, D. G., "Real-time UAV Trajectory Prediction for Safety Monitoring in Low-Altitude Airspace," *Aviation Forum 2019*, American Institute of Aeronautics and Astronautics, 2019.

[28] Banerjee, P., Gorospe, G., and Ancel, E., "3D Representation of UAV-obstacle Collision Risk Under Off-nominal Conditions," *2021 IEEE Aerospace Conference*, 2021, pp. 1–7. doi:10.1109/AERO50100.2021.9438182.

[29] Dill, E. T., Gilabert, R. V., and Young, S. S., “Safeguard – Flight Test Results of an On-board System Designed to Assure Conformance to Geospatial Limitations,” *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, London, UK, 2018, pp. 1–8. doi:10.1109/DASC.2018.8569289.

[30] Dill, E. T., Hayhurst, K. J., Young, S. D., and Narkawicz, A. J., “UAS Hazard Mitigation through Assured Compliance with Conformance Criteria,” *2018 AIAA Information Systems-AIAA Infotech @ Aerospace*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-1218.

[31] National Aeronautics and Space Administration, “NASA Software Engineering Requirements (NPR 7150.2D),” URL: https://nodis3.gsfc.nasa.gov/displayDir.cfm?t=NPR&c=7150&s=2B, 2022, [retrieved on 12 March 2022].