Using Fault Trees to Compute UAV Mission Risk

Abstract
Project Altiscope is using fault trees to better understand the factors driving UAS loss of control accidents. Through extensive modeling and statistical analysis, we find that weather, electrical system and maintenance-related variables have the greatest influence on whether a UAV is likely to lose control and crash. In modeling flyaway events, we note that communications link degradation and compass errors are the most significant predictors of a loss of control. Additionally, there is a need for greater training and certification standards for any people involved in the operation — even in the case of a fully autonomous fleet — to reduce the risk of human error. And we conclude that detect-and-avoid commands need to provide sufficient lead time for the vehicle to be able to react and remain within its performance envelope.

Introduction
As part of the risk framework development process, Project Altiscope has developed a pair of fault tree models to encompass a depth and breadth of UAV failure modes. This is an important step in identifying which underlying factors are most likely to contribute to a UAV loss of control or collision event. With that information, we can target our research and data collection efforts, which will give Altiscope's risk framework specificity and shape.

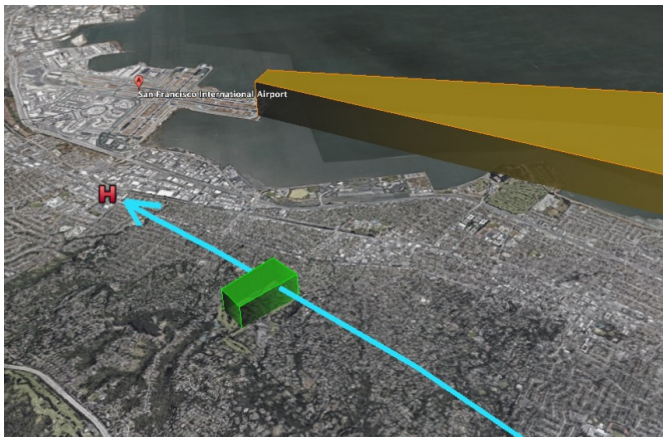
Our results are largely consistent with previous work done by others using fault trees to evaluate UAV mission risk. We find that weather factors; navigation reliability; operator error; and battery performance are the categories of events most in need of mitigation to reduce the frequency and severity of UAV safety events.

This paper presents our work in developing the fault trees and the results of sensitivity analyses performed on both trees. The analyses are especially important because we do not have data on exact component failure rates for each UAV model — factors that will vary by vehicle design, usage and environmental conditions. But we can test the relative significance of each event by applying a wide range of failure rates to each one, measuring how much of an influence those changes have on the probability of a top-level failure. We use these results in our conclusion to prioritize those events for which we need to gather data and conduct more research. These findings are also helping us determine which factors to focus on as we gather data and construct our risk framework.

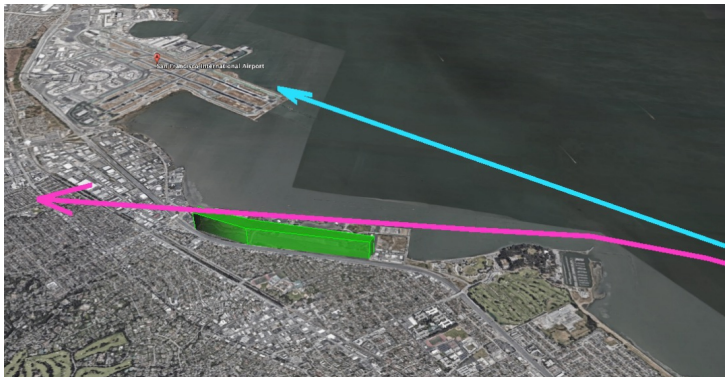
Notional Mission Profiles and Failure Risks
Our scope began with a relatively specific notional profile for a mission that might occur in the not-too-distant future: that of a small UAV with present-day capabilities doing photography or surveying work within a geofence region near a Class B airport. This made sense as a starting point because it's precisely what the Federal Aviation

Administration is attempting to address through its Part 107 airspace authorization request process -- although we envision future provisions that allow for more extensive, tailored use of controlled airspace (e.g. closer to the airport and at higher altitudes). This scenario forced us to immediately tackle a variety of complex interactions in developing the model. We assumed that frequently used commercial aircraft arrival and departure corridors are known, and that a system for handling UAV flight plans automatically rejects proposals for geofence volumes within or immediately adjacent to those corridors. The top-level event was a geofence escape resulting in either a mid-air collision, or a crash resulting in injuries or damage on the ground.

We soon discovered that a number of factors related to lack of collision/obstacle avoidance capabilities (or their failure) would contribute to a top-level event occurrence. Consider the following scenarios:



- The UAV is operating correctly within its geofence volume at an altitude of 1,000 feet AGL, which the regulator has determined to be an acceptable height given the site location in relation to the airport. However, a VFR helicopter is transitioning the Class B airspace on a random point-to-point route that intersects the geofence volume, creating a conflict about which air traffic controllers may be unaware.



- In a similar scenario as above, with the UAV operating correctly, an aircraft on arrival experiences an unrelated TCAS RA (Traffic Collision Avoidance System resolution advisory). The pilots decide to execute the published missed approach but are unaware that the charted procedure places the aircraft in conflict with the UAV.
- A local utility is surveying high-tension power lines that are offset from, but parallel to, the airport's primary arrival runway.
    - Due to a gust of wind, one of the UAV's mast arms impacts the top of a transmission tower, causing it to fly with limited control. The UAV flies out of the geofence volume and into the adjacent arrival corridor.

- Alternatively, the UAV encounters interference from nearby radio transmitter. This interference causes a communications link failure and anomalous compass readings. The UAV attempts to follow a correctly programmed return-to-home routine, but because of the compass error flies in the wrong direction, out of the geofence volume and into the arrival corridor.

Consideration of scenarios such as these led us to refine the model and develop two independent fault trees. We assume that basic mitigation measures are not followed, as this allows us to arrive at some "worst case" conclusions. In these examples, obvious mitigations may include restricting UAV position and altitude in proximity to commonly used arrival and departure routes; or terminating operations when winds approach a vehicle's performance limits. Vehicles may also have more advanced and redundant systems that reduce probability of erroneous navigation or compass guidance.

A UAV may experience a set of failures that lead to an unrecoverable loss of control, and subsequent crash or collision, as in the third example above. But that same vehicle may also encounter failures or environmental factors resulting in a controlled crash or collision — not unlike the factors that often contribute to controlled flight into terrain (CFIT) accidents in manned aircraft. Many of the basic events leading up to either a controlled or uncontrolled accident are independent from one another. As a result, a UAV could experience failures leading to one type of accident, independent of the probability of the other type. Thus, it made sense to model these as two different fault trees.

Fault Tree Background and Design Considerations

The fault tree analysis process is well documented in aerospace applications, as it enables systematic identification and review of the relationships between system components and their failure modes [3]. Fault trees are one way of visualizing the relationships between factors that increase safety risk and are often complementary to bow-tie analyses or Bayesian Belief Networks. Fault trees are advantageous because they provide a way to test failure probabilities without the steep learning curve of Bayesian Belief Networks, whereas bow-tie analyses are most commonly conducted in qualitative contexts. All three depict cause-and-effect relationships between combinations of threats (basic events on a fault tree), consequences (gates) and hazards (the top-level event on a fault tree). Just as each bow-tie diagram considers the factors surrounding a single hazard at the center, each fault tree uses combinations of basic events and gates arranged hierarchically and culminating in a single top-level event.

The most relevant literature we found is a paper describing a generic UAV fault tree [1] for beyond visual line of sight (BVLOS) operations. We discovered this paper after completing a large portion of our fault tree design work, and noted that we had independently arrived at similar design relationships, particularly for an uncontrolled crash or collision (Hammer's tree uses an equivalent top-level event). In response to Hammer's design and sensitivity analysis work, we adopted the additional "probability of loss of control" event to several second-level failures. However, where Hammer

models a robust branch related to obstacle detection failures, we envision an even more intricate set of relationships represented in the controlled crash/collision tree.

The fault trees are useful not only in articulating failure modes, but also in helping us identify exactly which kinds of data we need in order to generate meaningful failure probabilities. Some of that data, such as a battery's average failure rate, is specific to each battery model and likely proprietary to each manufacturer. Environmental factors such as wind and icing conditions are spatial and temporal, varying widely between nearby locations and over the course of a day (or even an hour). And we anticipate that autonomous drone traffic management systems that don't exist today will make errors in deconflicting two vehicles.

Our model only considers battery-powered multi-rotor vehicles, though many of the calculations can be extended to fixed-wing electric vehicles, or those with hybrid propulsion systems. The trees are designed to capture present-day Remotely Piloted Aircraft Systems (RPAS) as well as systems that may function with greater levels of autonomy, including BVLOS.

The first fault tree involves failures leading to a loss of control that results in a crash, collision or near-midair collision. All three are represented by the same top-level event, since a given sequence of failures may lead to any of them occurring depending on the UAV's location in relation to other aircraft and ground-based risk factors (e.g. critical infrastructure, large groups of people, etc.). The aim is to mitigate the risk of substantial damage, serious injury or death, which could occur to a passenger on the UAV, another aircraft or a bystander on the ground. In this first tree, failures in any one of four branches (described below) can lead to the loss of control. That is, neither a human operator nor an autonomous management system would be able to regain control of the vehicle to make a safe diversionary landing.

The second fault tree evaluates more complex system failures resulting in the same top-level event as described above. Modeling the same top-level event in both trees is important so that we can draw comparisons between them. These failures may include equipment malfunctions that don't immediately result in a loss of control, such as the failure of an onboard obstacle sensor. But in combination with other failures in the tree, this could result in a vehicle that encounters a crash or collision scenario even though its flight controls, navigation and propulsion systems were working correctly.

Fault Tree 1: Losses of Control

Similar to Hammer's approach, in designing this tree (Figure A-1) we posit that a failure in any one of four branches -- hardware, weather, maintenance or flyaway scenarios -- can result in an unrecoverable loss of control and crash, independent of the other three branches. Full depictions of each branch, including mean unavailability assigned to each event, are located in Figures A-2 through A-5. Basic events PLC1 through PLC4 are fixed probabilities to address the fact that a vehicle may remain controllable (or at least able to make a forced landing) in some circumstances after a failure in the respective branch.

Starting from the left, the first branch considers a wide variety of vehicle hardware failures (Figure A-2). These are broadly categorized as propulsion, battery and electrical system failures. We did not model hybrid vehicle power designs that use a gas turbine engine to generate electricity for the motors and other systems. Those design complexities will require additional modeling for evaluating mission risk in vehicles such as the prototype SureFly Workhorse. The hardware branch also includes failure of the vehicle's electronic speed controller and the flight computer; as well as multiple motor, rotor and wiring failure modes. The tree takes a very conservative approach in assuming that the failure of one motor on a quadcopter (or two motors on a multirotor vehicle with six or eight powerplants) constitutes a failure that could propagate to a loss of vehicle control. Actual design considerations, including software control routines unique to each vehicle model, may allow some vehicles to continue flight, but with limitations on range, altitude, thrust or degrees of freedom (i.e. maneuverability).

The flyaway branch (Figure A-3) is of particular interest, because these incidents are often the result of compound failures influenced by environmental factors. We modeled this branch with four sub-parts:

- Compass error, which could result from a failure to properly calibrate it; electromagnetic interference (EMI); or the actual failure of the onboard solid-state compass.
- Communications failure or latency (link loss) resulting from either the failure of the onboard radio or a remote station; or signal degradation between the vehicle and ground station.
- An unreachable home point (or rally/ditching location) due to winds aloft being higher than predicted, and the point being into the wind relative to the vehicle location (the vehicle may be unable to reach the home point due to winds that exceed available forward thrust; or due to battery range limits).
- A GNSS failure due to signal loss (modeled separately due to EMI or poor satellite coverage) or the failure of either the onboard GPS antenna or receiver.

A failure of any one of these four subparts does not result in a crash because of the semi-redundant nature of most vehicle systems. For example, without GPS, the vehicle can still use the IMU and compass to dead reckon and follow an operator's commands to manually land. Without a communications link, the vehicle can use its GPS and compass to navigate itself to the home point and land. To represent these dependencies, we use a voting gate above all four elements: any two (or more) of the four must fail at the same time. This would mean, for example, a situation in which a vehicle loses GPS and communications link, rendering it unable to find its way to the home point and the operator unable to send a "land now" command. The vehicle's behavior may be unpredictable, depending on how the manufacturer configured it to behave and whether the user correctly set the home point and calibrated the compass before flight. If it regains communications link or GPS, then it can attempt to land. Such a recovery is broadly represented by PLC2, a fixed probability of loss of control.

The third branch (Figure A-4) of the tree depicts four categories of weather conditions that may cause a loss of UAV control: high winds at the takeoff or landing site; icing conditions; low visibility; and convective activity. We do not include inflight winds because a strong headwind condition is already represented in the flyaway branch. Low visibility is included here both because line-of-sight is a requirement for many present-day operations; and because vehicles may use optical sensors both to gauge position when close to the ground, and to detect obstacles.

Finally, the fourth branch (Figure A-5) includes failures resulting from maintenance actions and inflight structural impact. This includes human error in failing to follow correct maintenance procedures, and in failing to identify loose or missing parts that could result in an inflight breakup.

Fault Tree 2: Redundant System Failures

With the exception of several weather factors, this tree (Figure A-6 through A-9) is completely unique compared to the previous. Whereas the previous tree includes both temporal threats and failures that are the result of fatigue or random error, this tree models the breakdown in redundancies that may occur, especially in BVLOS or fully autonomous regimes. The tree assumes that the vehicle is equipped with onboard sensors to detect and avoid other traffic or obstacles. It also assumes the vehicle has a link with some type of service that provides control and routing instructions to avoid vehicle conflicts. This could include present-day capabilities available to users of some third-party fleet management software, or future UAS traffic management (UTM) systems.

The top level event is a voting gate, in which two of the three gates below it must fail at the same time:

- **Route Conflict (Figure A-7).** Two vehicles are in conflict with each other because a traffic management system isn't aware of one of the vehicles; or one vehicle unexpectedly deviated from a previously approved course. Deviations may occur due to inflight emergency, unexpected weather or a change in destination. Vehicles can either conflict in crossing/head-on trajectories; or one may be overtaking the other because of different cruise speeds. A fixed probability variable in this branch controls for the obvious fact that both vehicles must be at the same altitude (even if one is descending or climbing into the path of another).
- **Human-in-the-Loop (HITL) (Figure A-8).** Human error may take a variety of forms in UAV operations, even when the flight is considered to be fully autonomous. In VLOS and BVLOS settings, the operator may lack sufficient training, experience or good judgment to ensure the safety of the flight. Preflight errors, such as failing to follow correct cargo loading procedures or calculating flight plan information incorrectly, may also jeopardize the success of the flight. Or an operator may intentionally try to crash the vehicle to cause harm to other people. In environments with increasing levels of autonomy, a remote operator

overseeing multiple vehicles may not react quickly enough to correct a deviation or avoid a conflict.

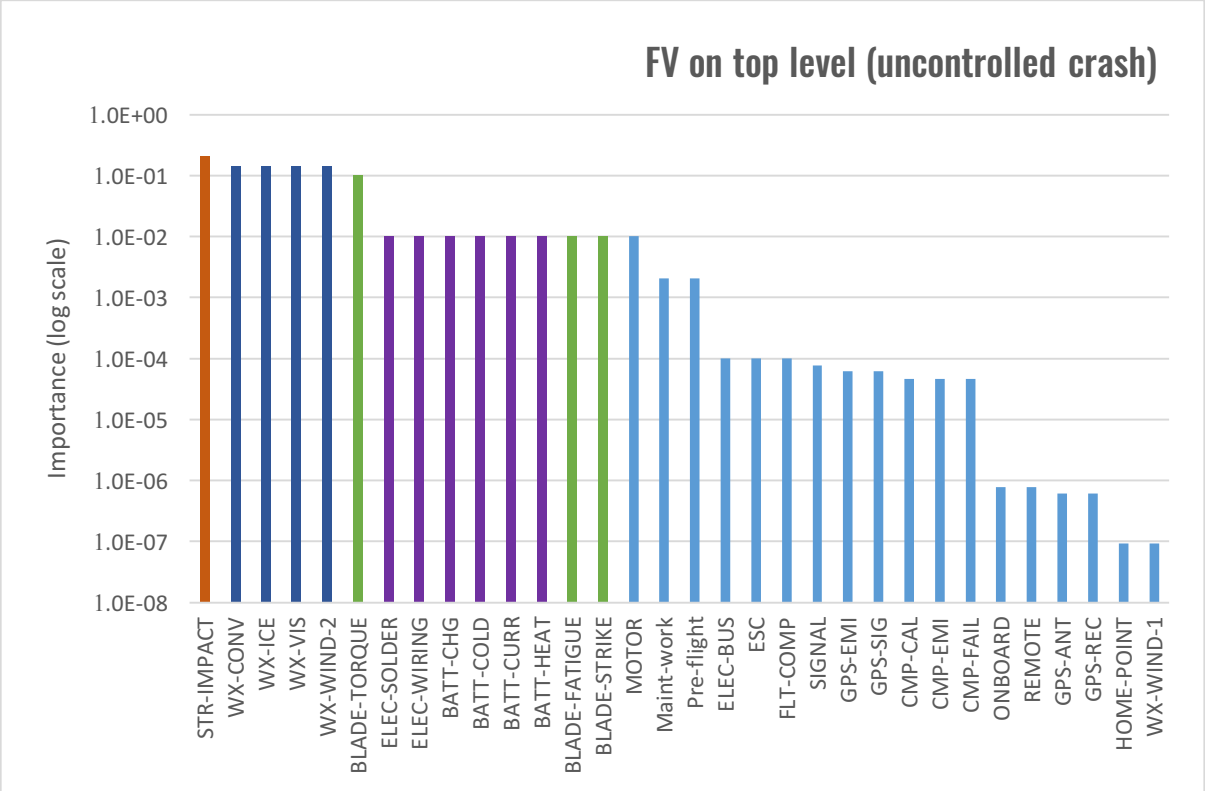
- **Deconfliction Failure (Figure A-9).** The third branch includes multiple pieces of equipment a vehicle may have to help it avoid obstacles and other traffic, including a transponder (specific mode/protocol was not considered), optical sensors and LIDAR. Those components could fail, making it impossible for the vehicle to detect inflight hazards. Alternatively, while that equipment may be working correctly, the avoidance command may come too late, requiring climbs, turns or other maneuvers that would exceed the vehicle's performance envelope.

Sensitivity Analysis Methodology

We designed both fault trees using the Relyence web app interface, treating the failure of each basic event as mean unavailability. We tried to make reasonable assumptions in assigning mean unavailability values, but we are cognizant that the mean values assigned have a significant role in influencing higher-level unavailability. Because of this variability and uncertainty, we calculated several different statistical indicators of uncertainty and importance. These include:

- Fussell-Vesely (FV), a formula which measures an event's overall contribution to higher-level risk probability. With values between 0 and 1, lower values indicate an event has negligible influence on failure rates, while higher values are more critical to system reliability.
- Risk Achievement Worth (RAW), which measures the increase in top-level event failure if a given basic event's mean unavailability is set to 1 (that is, always unavailable).
- Risk Reduction Worth (RRW), which indicates the reduction in top-level failure if a basic event's mean unavailability is set to 0 (that is, always available).
- Birnbaum's Importance (BI), is a derivative function that measures the rate of change in system risk as the probability of a basic event's failure changes. BI is equivalent to the sum of RAW and RRW, and therefore is not influenced by the mean unavailability of a basic event.

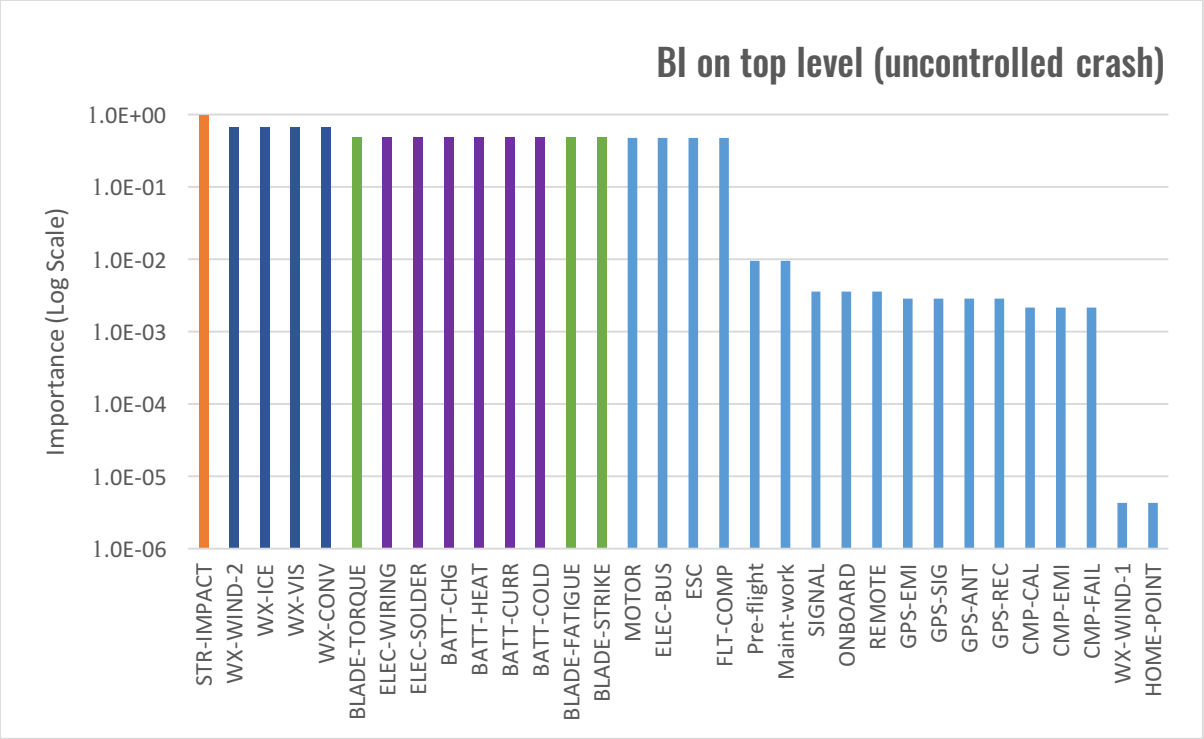
The above measures are calculated using different formulas, and examining multiple measures may be useful when addressing varying objectives [2]. FV is commonly used to rank basic events based on their importance on higher-level outcomes, but it can be sensitive to small baseline probabilities. BI is also useful as it reveals the sensitivity of top-level risk to changes in basic event probabilities. We examined FV and BI for basic events on both the top-level as well as the branches (hardware, flyaway, weather, maintenance). The branch sensitivity measures provide insight into how basic events affect particular modes of failure.



We used the Relyence web application to create and export the fault trees. Next, we analyzed the fault trees using the statistical software R to conduct the sensitivity analysis. While Relyence automatically calculates similar measures for FV (Diagnostic) and BI (Marginal), at the time of writing the company provided limited documentation on the formulas and calculation process behind those measures. Therefore, we selected R because of its ability to run custom reproducible simulations and its robust documentation of methodology.

Sensitivity Analysis Findings

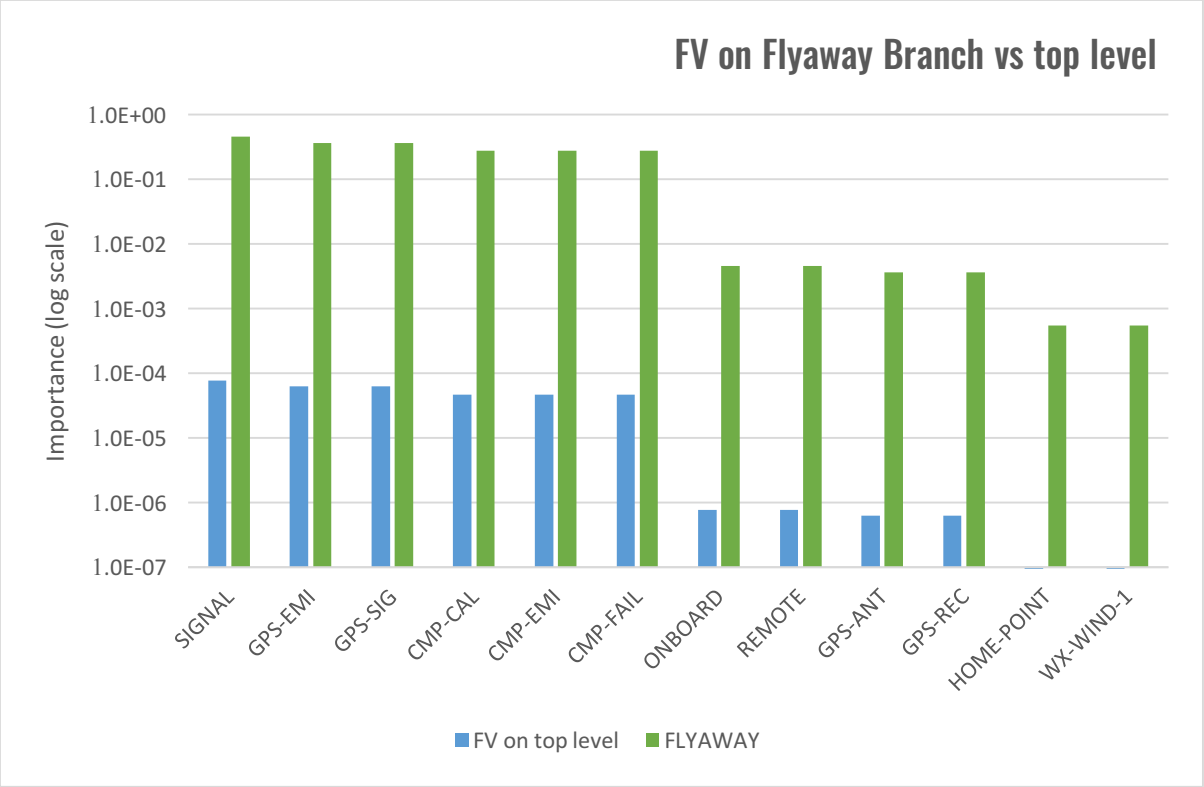
The charts below show the FV and BI values of each event in Fault Tree 1 (loss of control) plotted on a log scale. The events are very closely correlated in importance using both measures. Most significant is an impact that damages a vehicle's mast arm or rotor assembly (STR-IMPACT FV 0.21, BI 0.96). This event is located in the maintenance branch (Figure 5) one level above two other events, all of which were assigned mean unavailability of 0.01.



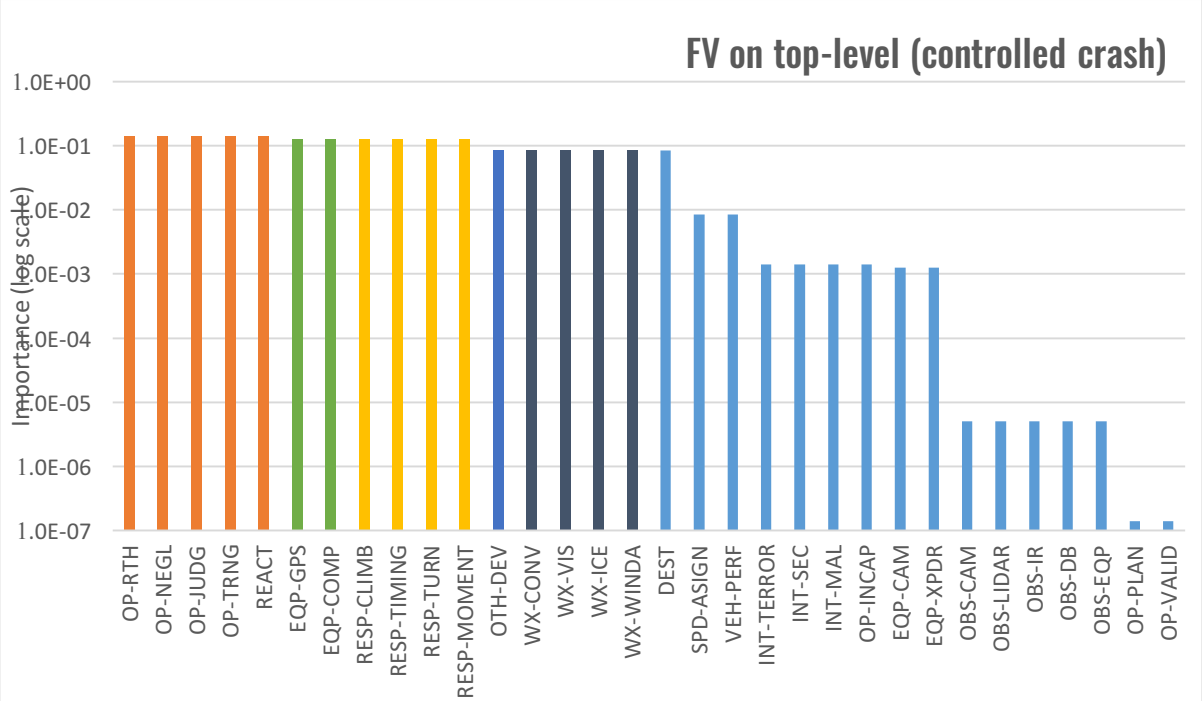
All four weather events are the next most important (FV 0.14, BI 0.67). All are at the same level in the tree depth and also assigned mean unavailability of 0.01. Of course, weather is temporal and can vary between locations less than a kilometer apart. Current high-resolution weather models provide short-term predictions at a 1-km scale, which would supersede the mean unavailability values.

Six events for various battery and electrical issues also carry high importance values (FV 0.01, BI 0.47), but notably an order of magnitude lower when expressed in FV.

We also performed sensitivity analysis calculations for each event on the second-level failures (flyaway, hardware, maintenance and weather). The chart at right compares the FV values for the basic events in the flyaway branch on the top level of the entire tree (blue) versus on just that branch (green).

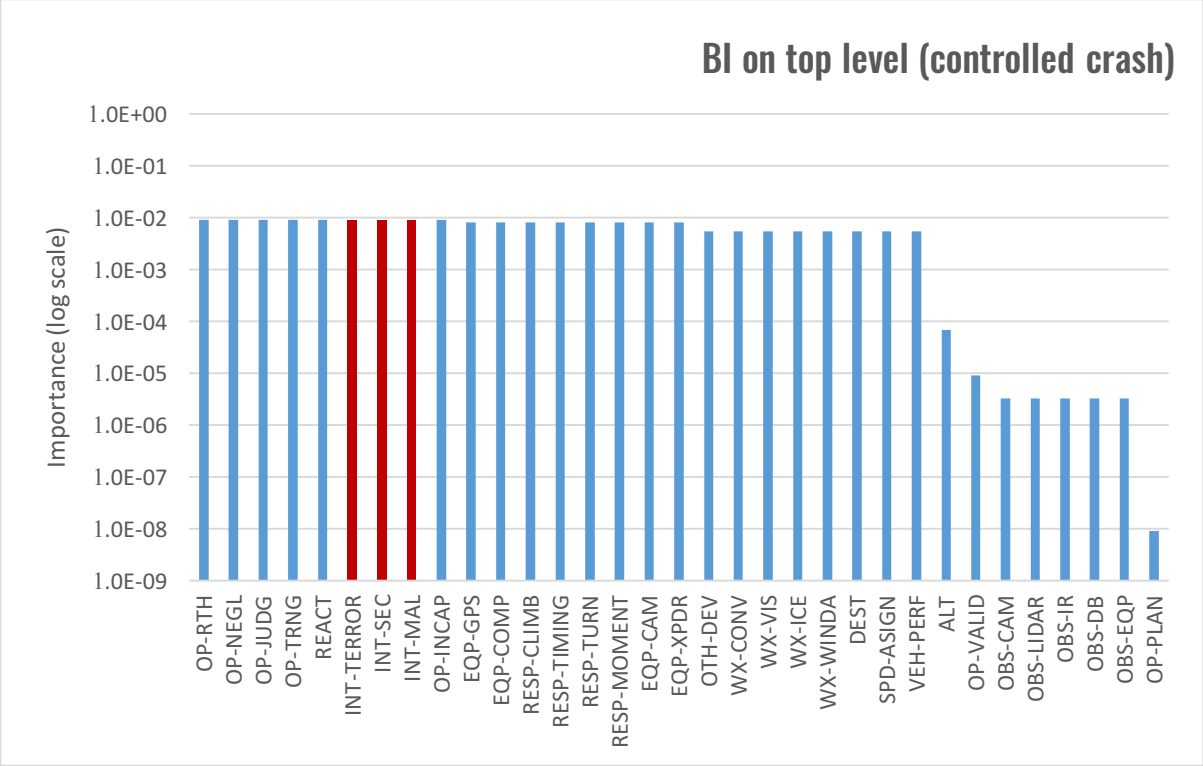


The results show that while compass and GPS reliability issues carry an importance of between 4.7E-5 (three events related to compass failure and inaccuracy) and 7.7E-5 (communications link loss or degradation) on the top of the tree, they are much more impactful within their branch, with FVs that rise in importance to between 0.28 and 0.45, respectively.



The analysis of the second fault tree (system redundancies leading to a controlled crash) found that five events related to operator training, skill and judgment had the largest FV values, all 0.14. Two equipment failures, that of the onboard GPS and

compass, shared the next-highest FV values, at 0.13. In addition, four inputs on the right side of the tree related to how quickly or aggressively a vehicle could maneuver in response to an obstacle avoidance command shared an FV value at 0.13.



These values are comparatively low because the top-level event only occurs if failures in two of the three branches occur at the same time. Because of that dynamic, the BI measures are all much lower, as one might expect. No single event leads to a crash or collision, and conversely, mitigating a single event cannot substantially reduce that risk. Most of the BI values correlate to FV values for a given event, with the noteworthy exception of three events related to intentional efforts by a person to cause a crash (acts of terrorism, cybersecurity threats and other malicious intent).

Conclusions

In Fault Tree 1, we found that weather, electrical system and maintenance-related variables have the greatest influence on whether a UAV is likely to lose control and crash. This is consistent with Hammer's findings, and we arrive at a similar conclusion that systematic maintenance procedures and accurate weather data are important in mitigating that risk. In evaluating the flyaway sub-branch, we find that communications link degradation and compass errors are the most significant contributors. This suggests that ensuring highly reliable onboard equipment alone is not enough to reduce flyaway risk — we must also find ways to mitigate against signal loss.

Fault Tree 2 provides additional areas for emphasis: the need for training and certification standards for any people involved, whether they are remote pilots or monitoring an otherwise autonomous fleet. And we can see that detect-and-avoid commands need to come with sufficient notice for the vehicle to be able to react.

Our risk framework (and therefore any model implementations) needs to focus on the following elements because of their overall influence in affecting the successful outcome of a flight:

- Access to location-specific real-time weather data sources in combination with a robust database of vehicle performance attributes and operating environment tolerances.
- One or more input variables to represent the vehicle's maintenance status and the quality of repair work (for example, were repairs done by a certified mechanic and tested in accordance with future industry or regulatory standards?)
- At least one input variable indicating the operator's qualifications, certifications and experience level.
- A data layer representing ground-based sources of EMI, which might include electrical infrastructure; cellular and radio transmission towers; and known localized areas of geomagnetic disturbance.
- Enabling regulatory and industry consensus on clear definitions for closest allowable proximity between vehicles (whether separation standards in airspace managed by air traffic controllers; an automated Drone Traffic Management system, or "well-clear" detect and avoid thresholds) and sensor arrays that can detect conflicts with sufficient advance warning time.

Acknowledgements
Chris and Erin Dienes made substantial contributions to this paper, including conducting several iterations of the sensitivity analysis, highlighting structural issues in early versions of the fault trees and providing careful documentation of their methodology and findings.

Project Altiscope would like to thank Mark Dombroff, Simon Hennin, Rob Knochenhauer, Peng Wei and Steve Weidner for providing critique and feedback on drafts of this paper.

References

[1] Hammer, J. et al. 2017. Safety analysis paradigm for UAS: Development and use of a common architecture and fault tree model. *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)* (2017), 1–10.
[2] Idaho National Laboratory Importance Measures.
[3] Stamatelatos, M. et al. 2002. *Fault Tree Handbook with Aerospace Applications*. NASA Headquarters Office of Safety and Mission Assurance.

Annex A:
Fault Tree Diagrams

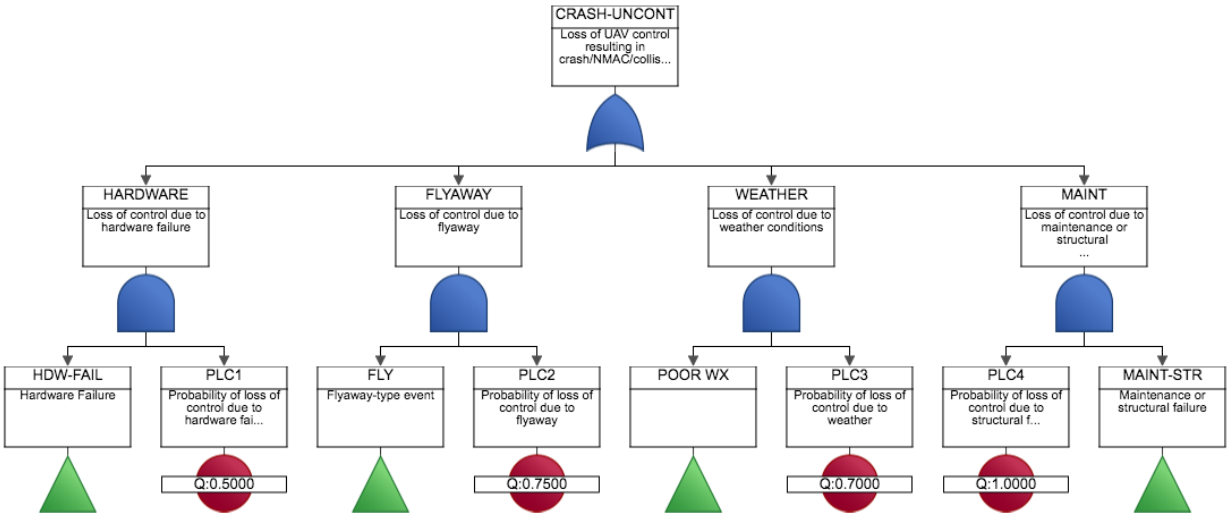


Figure A-1. Uncontrolled crash, collision or near-midair event. Green symbols indicate branches depicted in subsequent figures.

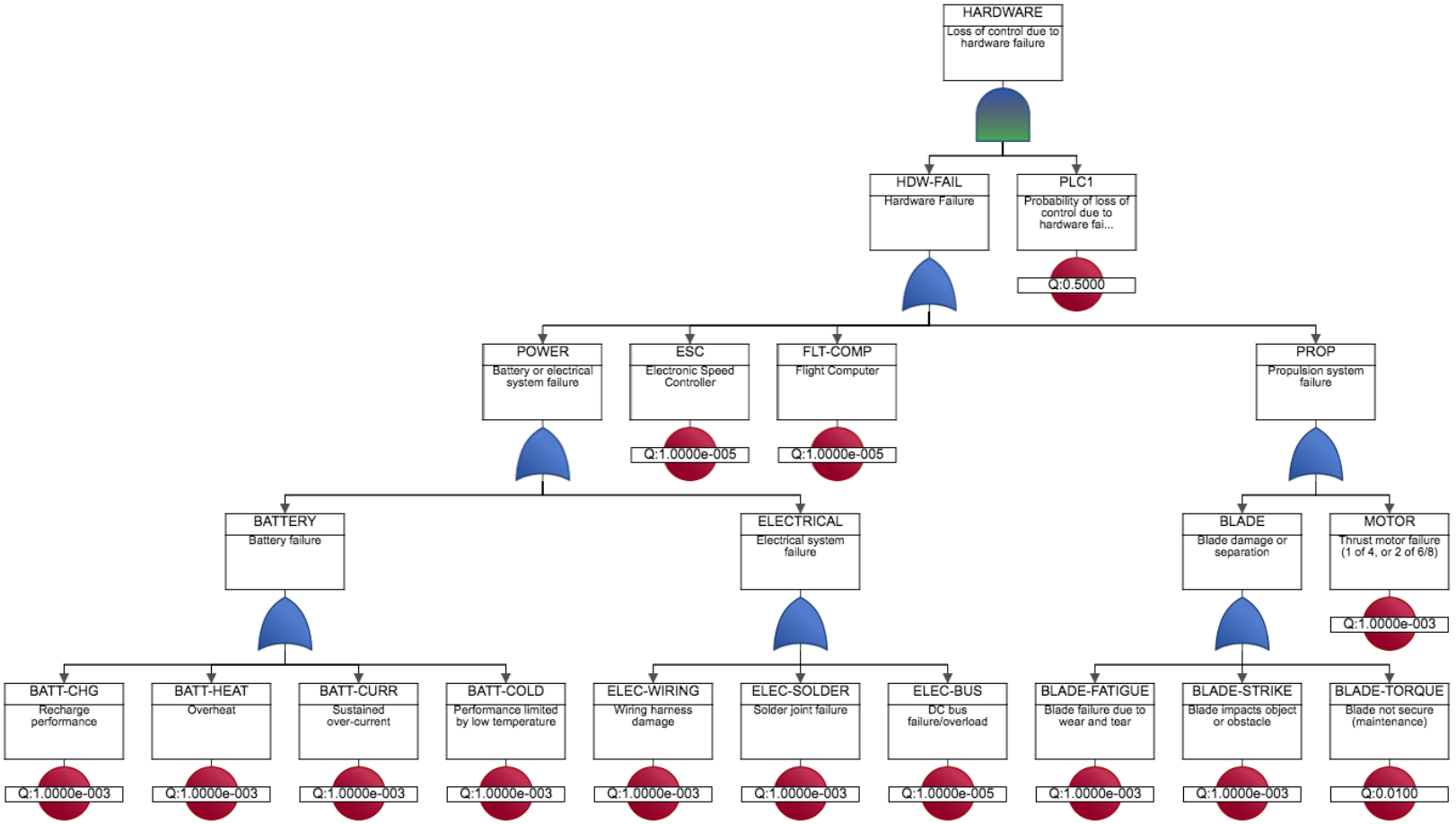


Figure A-2. Hardware failure branch with mean unavailability rates used in sensitivity analyses.

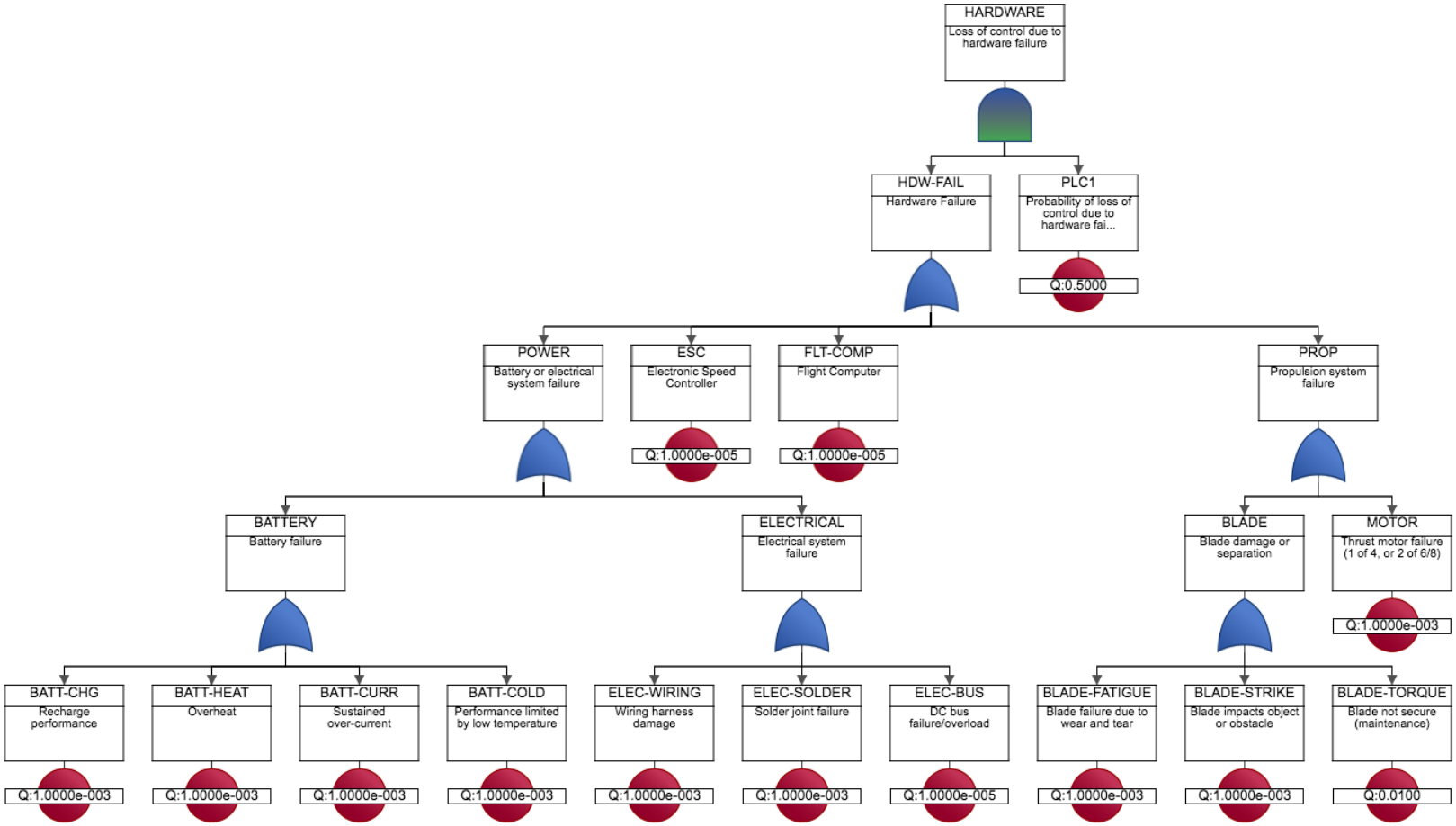


Figure A-2. Hardware failure branch with mean unavailability rates used in sensitivity analyses.

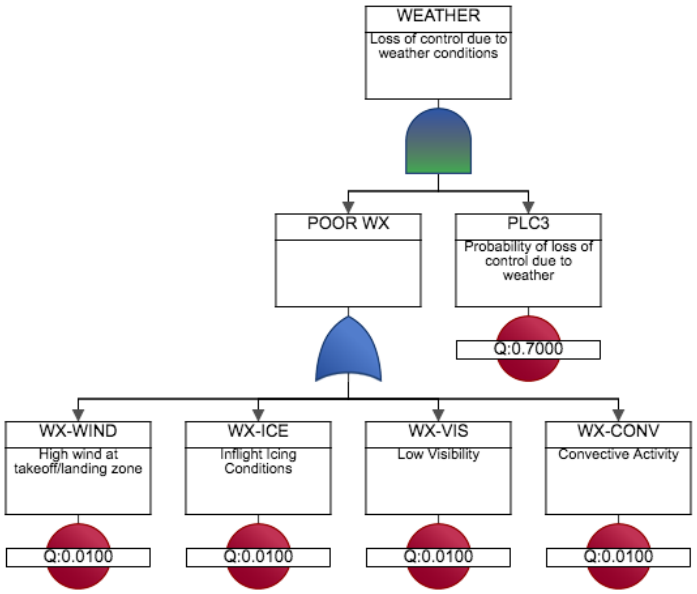


Figure A-4. Weather failure branch with mean unavailability rates used in sensitivity analyses.

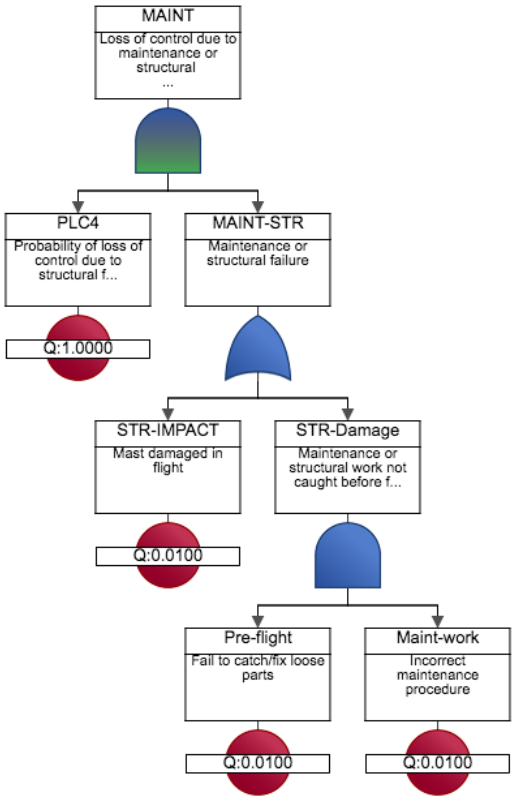


Figure A-5. Maintenance failure branch with mean unavailability rates used in sensitivity analyses.

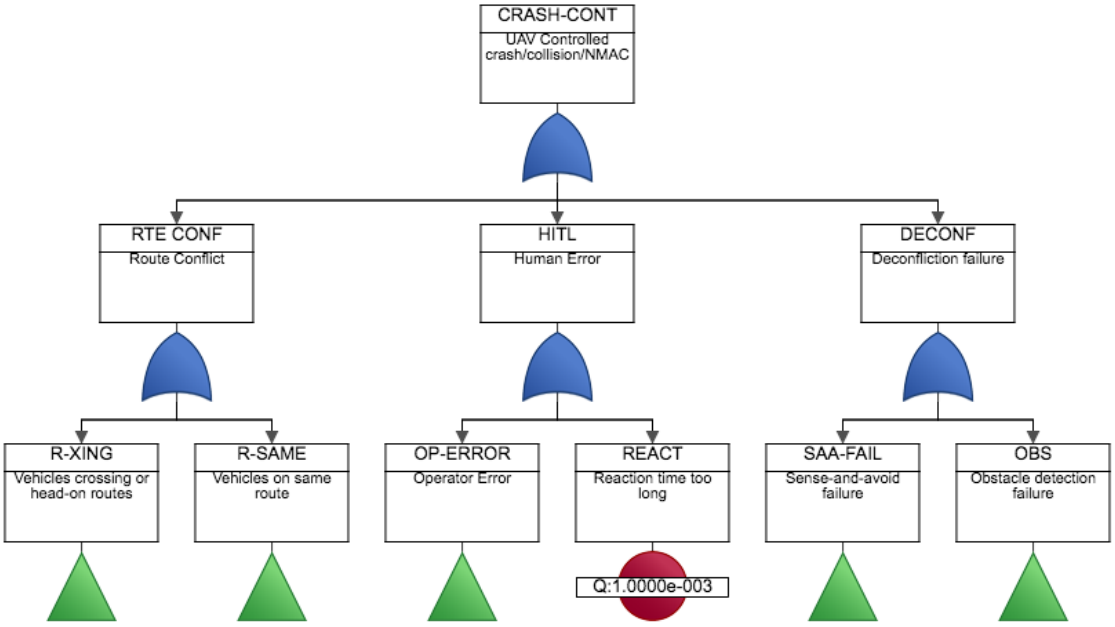


Figure A-6. Controlled crash, collision or near-midair event resulting from redundant system failures or human error. Green symbols indicate branches depicted in subsequent figures.

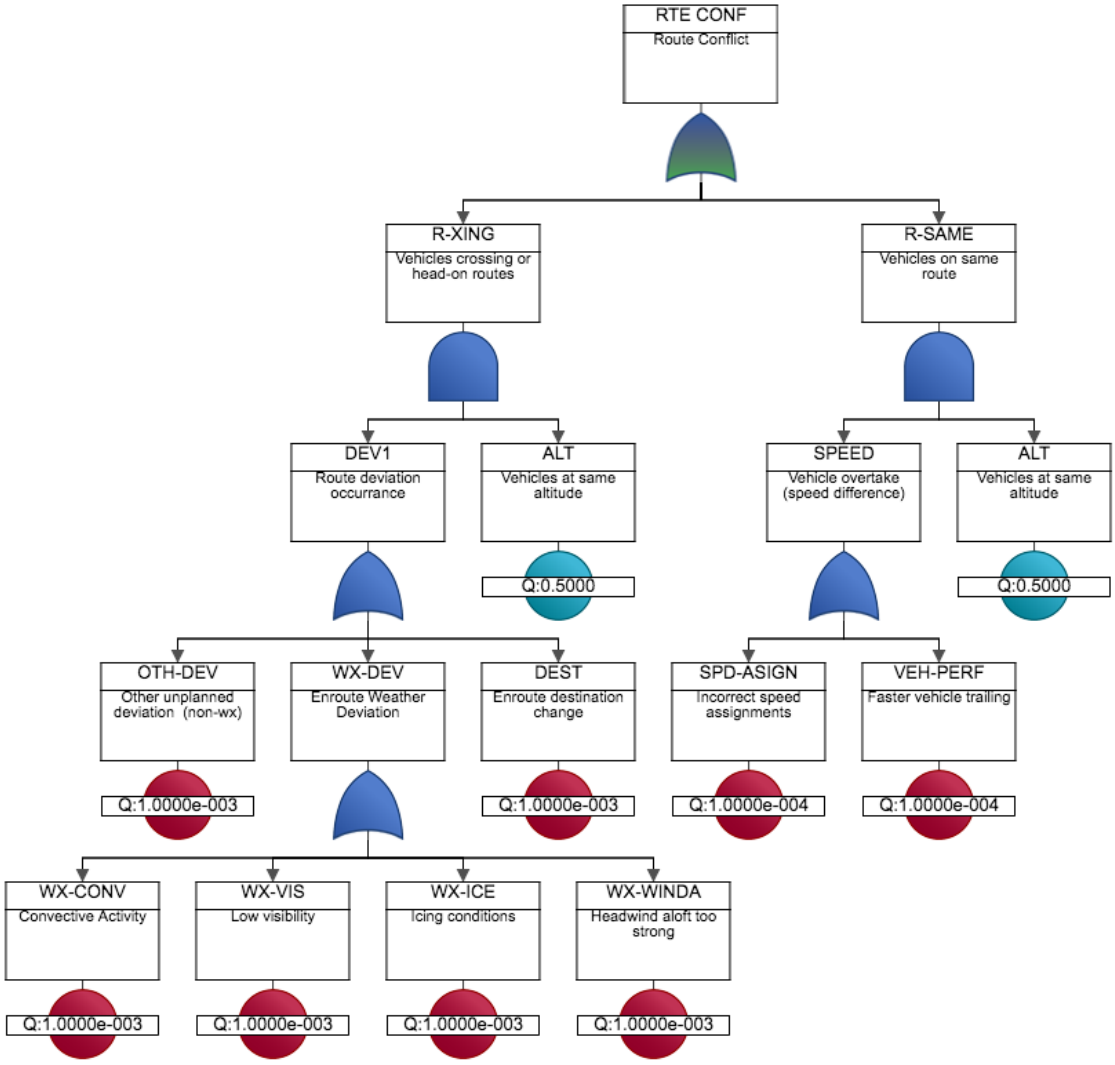


Figure A-7. Route conflict failure branch with mean unavailability rates used in sensitivity analyses.

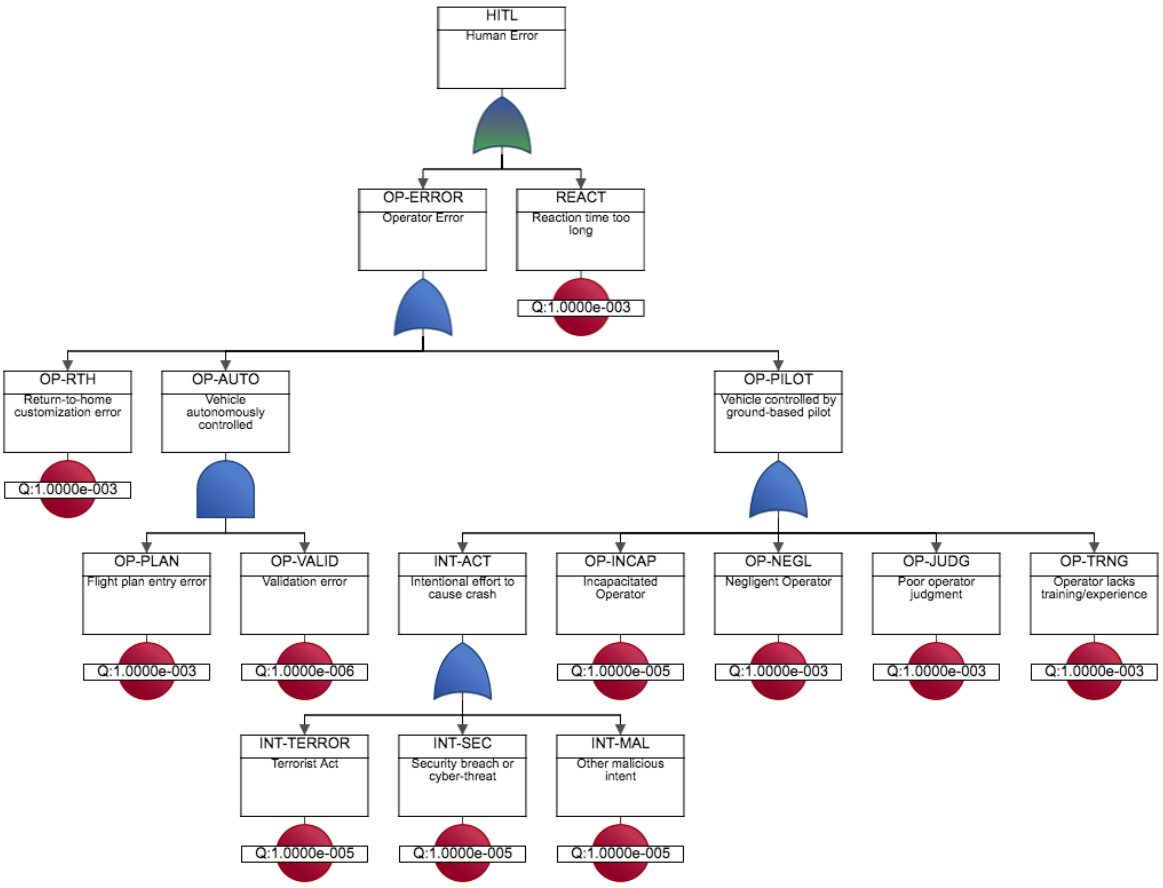


Figure A-8. Human error failure branch with mean unavailability rates used in sensitivity analyses.

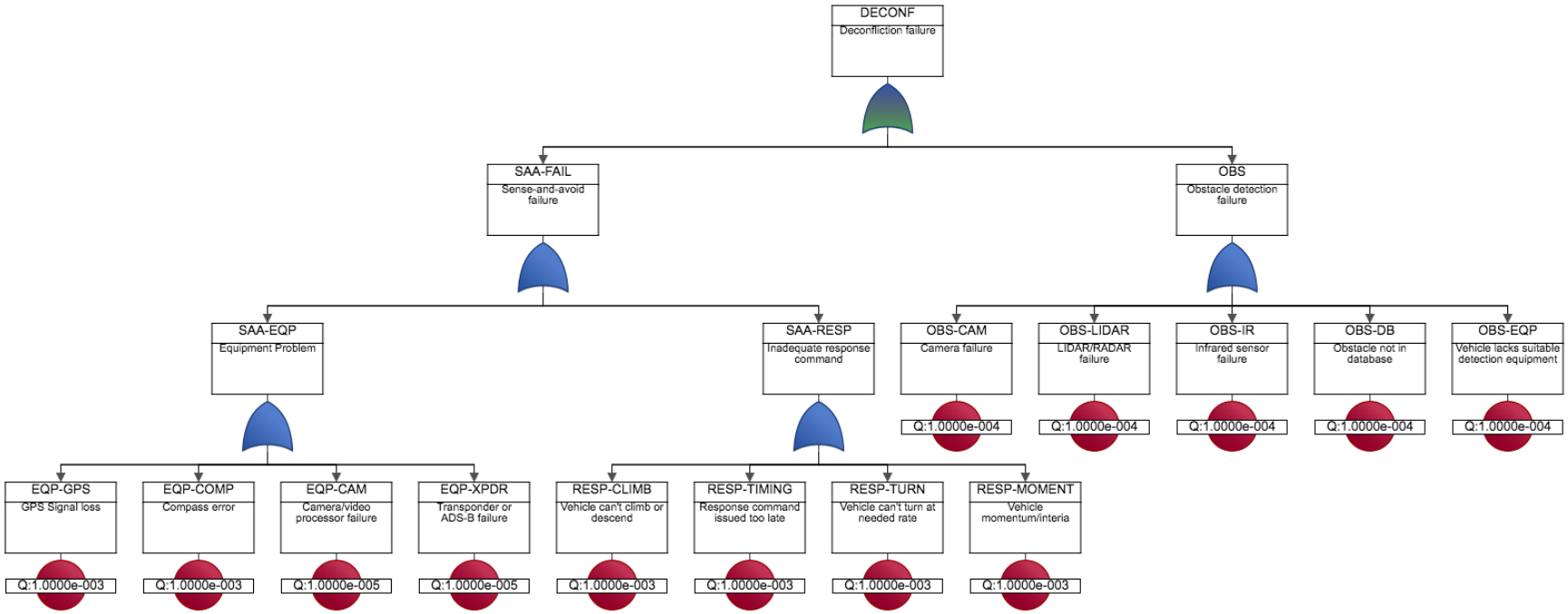


Figure A-9. Deconfliction failure branch with mean unavailability rates used in sensitivity analyses.