Leveraging ASTM Industry Standard F3269-17 for Providing Safe Operations of a Highly Autonomous Aircraft

Mark A, Skoog

Principal Investigator for Autonomy

NASA Armstrong Flight Research Center

Loyd Hook

Department of Electrical and Computer Engineering

University of Tulsa

Wes Ryan

Small Aircraft Directorate

Federal Aviation Administration

Submission due 18 Oct 2019

Revision due Jan 2020

Abstract

This paper expands upon the ASTM industry standard F3269-17 to outline a run-time assurance (RTA) network architecture for use in ensuring safe flight operations of a highly autonomous aircraft. An RTA network architecture is proposed and critical features discussed to implement functions where automation is primarily responsible for the safety of the aircraft instead of a pilot. This shift in responsibility, made possible by the proposed architecture, is key to highly resilient automation and is a core enabler for future “pilotless” transportation concepts. The findings in this paper stem from the

researcher’s experiences with ASTM in the generation of the standard and some seven years of RTA system development on various flight programs leveraging the RTA concepts outlined in the ASTM standard.

Introduction

A major part of the twentieth century was dominated by tremendous advances in transportation and technology. Beginning with the horse and buggy and ending with transcontinental super highways and air transportation, technology has enabled intermodal trans-global travel in less than a day. However as the twenty first century has progressed, population increases in major metropolitan areas has outpaced the capacity of the highway system. This has led to an increase in door to door time for most of our population when traveling. Even with air transportation greatly improving the efficiency and access for air travel, we are still faced with the traffic getting to and from an airport.

Another major advancement has been in computer and automation technology, leading to the retail market brought on by the internet and highly-automated computer-controlled systems. The internet has increased access to goods purchased online, but for items purchased or delivered in the same day, there is still exposure to the growing traffic congestion mentioned above. Self-driving automobiles and trucks could offer a minor improvement to the situation, but this must be done safely and with resilient automation.

Another possible solution to these problems has been proposed by taking transportation to the skies in the form of urban air mobility and drone package delivery. By adding additional dimensions to available avenues of transportation, quicker access to retail and a continued shortening of travel times across major metropolitan areas would be allowed. However, currently all forms of air transportation require highly trained pilots to fly these vehicles to provide safe operation. When the additional cost of a highly trained pilot is added to these envisioned future operations the business case may be adversely affected. Thus, many purveyors of these futuristic concepts have been advocating for and working on autonomous and pilotless alternatives. However, in order to satisfy regulators and the public, these systems must provide a net gain in efficiency while ensuring no degradation in safety. However, this is not a trivial endeavor as exemplified by the twenty years of work developing standards for the sense and avoid requirement on unmanned aircraft. Further, dramatically increased complexity brought about by on-demand mobility to the surface in a dense urban environment, required of urban air mobility and package delivery, will only exacerbate this problem. However, only when this is achieved and demonstrated will highly trained operators not be required.

Thus it is becoming clear that new methods for design and testing will be required, along with new system architectures in order to enable fail-safe, fail-functional autonomous systems that are capable of continued safe flight and, if needed, landing without human intervention in off-nominal cases. RTA concepts may be a key enabler to this transition. They allow systems to be designed and monitored for proper function and nominal conditions and provide simple deterministic alternatives when conditional barriers are about to be breached [1], [2]. RTA concepts also allow for overarching performance/safety objectives to bound the behavior of the system, along with the behavior of individual functions. Unfortunately, current certification standards do not give credit for an RTA approach to safety; however, government and industry are working to change that[3].

NASA and the FAA have been collaborating on the development of new safety standards for safely operating unmanned aircraft for many years. In one of these collaborations, NASA is leveraging their experience gained from F-16 Automatic Ground Collision Avoidance System (Auto GCAS) development [4], [5] and other automatic safety systems to work with the FAA and ASTM to define a safety architecture that could be used to ensure safe flight operations of a highly autonomous aircraft. This collaboration contributed to the publishing of ASTM Industry Standard F3269-17 [6], [7] which uses an RTA architecture, shown in Figure 1, to safely bound a complex system.

The NASA team has also been developing and integrating an instantiation of such an architecture onto various small UAVs (sUAV) with the intention of exercising pilotless operation[8]. The term "pilotless" in this context refers to a vehicle with an onboard computerized system that is capable of sensing, evaluating and taking action by automatically maneuvering or adjusting flight plans to address numerous safety responsibilities that are typically the responsibility of the pilot in command. Most recently the DoD has joined the NASA / FAA collaboration under the Resilient Autonomy (RA) project. The RA project has a goal to expand upon F3269-17 to create an RTA network of safety functions that is intended to lead to a pilotless aircraft that can safely fly in select portions of our airspace.

Findings from the Resilient Autonomy project will be published over the course of the project to inform the community of possible ways to apply F3269 towards certified operation. This paper is the first such publication. Four major topics are covered within this paper, which is structured as follows. Section 2 discusses splitting the RTA "Monitor" into multiple distinctly separate monitors. Section 3 expands upon the ASTM "Input Manger" in addressing the integrity of the sensed values in the context of safe operations. Section 4 describes how failures detected by the "Input Manager" can be addressed. Section 5 notes how this architecture can be adapted to a preexisting aircraft autopilot with a minimum of integration and how that affects the "Trusted Controller." Lastly, the paper is concluded with a summary and future and envisioned work.

The Use of Multiple Separate Monitors

Historically safety systems have been developed to address a single aspect of safety. For instance, TAWS mitigates the potential of ground impact and TCAS mitigates the potential of a midair collision. During F-16 automatic collision avoidance development, it was the mutual consensus of the team that combining both air and ground collision avoidance algorithms into a single algorithm was far more complex than the sum of the individual complexity of the two algorithms. Similarly, as the Resilient Autonomy team looked at addressing even more safety responsibilities such as well clear, geo-boundaries and safe means to address emergency conditions that would require an emergency landing or safe ditch, the complexity of a single algorithm exploded.

A path forward was found by considering how pilots address emergency situations where time-critical decisions needed to be made. Experience, both from years of flight testing as well as the evaluation of many F-16 mishaps, showed that pilots will focus their attention on a single task when that task is very critical to safety. Therefore, it was decided to make each safety function a separate monitor within the RTA context. Each of these monitors is a separate software isolated module (run in parallel) rather than one single combined algorithm or monitor.

The software separation of each monitor allows each individual monitor to be mapped to a specific pilot responsibility, better ensuring that all intended safety responsibilities are being appropriately covered. Also, this design choice has the added critical effect of allowing each individual monitor to be more easily verified and validated than a single "mother of all" monitor. These factors combine to create a more robust and therefore trustworthy monitor. Although the "Safety Monitor" in Figure 1 could be considered to have multiple functions within the monitor, the Resilient Autonomy team felt it was important to explicitly call out the benefit of multiple separate monitors to people attempting to use the standard for pilotless operation.

A new issue arises when multiple separate monitors are implemented in an RTA architecture. This issue is that multiple switch positions will likely be required within the RTA Switch for the different control methodologies used to resolve the boundary breach that the different monitors may flag. The multiple switch positions are illustrated in Figure 1 however, the selection of which switch position that should be selected at any point in time is extremely critical to the safety case that must be made. Even more an issue is how to address switch position selection when multiple monitors are flagging boundary breaches at the same time. What methods should be used to adjudicate which recovery control function should be given control authority? The team resolved this by again looking to how pilots today successfully address multiple safety issues that are occurring at the same time. Pilots focus first on what they believe is the most critical issue in terms of near-term likelihood and potential outcome. Once resolved, they move on to the next most critical issue.

For implementation each monitor is then required to declare the criticality (or expected consequence) of the boundary breach that they are flagging. One method of quantifying consequence requires weighing monetary loss against appropriate behavior, human harm and loss of life. Therefore, the architecture should require each monitor to access rules of behavior quantifying the consequence of breaching its boundary and report that consequence to the “RTA Switch”.

It should be noted that design engineers should not be deciding the weighting between loss of life and mission success for these rules of behavior. Instead, society and our legal system should be setting these rules of behavior. Therefore, these rules may change from one geographic region to the next, such as governing privacy rights or in response to drastically different population densities. Therefore, the RTA network should allow different rules to be plugged in and provide a guarantee those rules will govern priorities on the aircraft.

The Vital Importance of Data and Flight Integrity

Within F3269 the functional requirements for the input manager are discussed. One of these requirements is to “monitor the integrity of the RTA required inputs and annunciate a failure if the input is corrupted”. No discussion is made of what should be done if a failure is detected nor is there any provision for addressing failures within Figure 1. Instead, the requirements for the Recovery Control Function call for each function to be able to handle failed conditions. The Resilient Autonomy team has given this considerable thought and believe that the delegation of handling all failed conditions to each recovery function leads to considerable duplication of functionality and adds considerable complexity under at least some failed conditions. The team's alternative approaches are presented here.

After the standard calls out the functional requirement to "monitor the integrity", it gives one additional derivative requirement regarding redundancy. It then goes on to list other functional requirements for the "Input Manager" such as range checking, stale data checks, etc. Redundancy is one of many methods for determining data integrity. Range checking, stale data checks are just a couple other examples of determining data integrity. Redundancy has been highly used to great success within the aviation community for determining the integrity of flight and safety critical data. Redundancy brings with it the penalty of cost, weight and complexity. Additionally, sensors being envisioned for driving monitor functions such as sense and avoid or obstacle avoidance require sensors that can be "blind" against certain threats under certain environmental conditions. These pose a common failure condition for redundant sensors and therefore conditions where no additional integrity is provided by redundancy.

The Resilient Autonomy team believes that a designer developing an Input Manager should place an emphasis on data integrity, even to the point that this part of the RTA diagram may better be labeled the "Integrity Manager". All available means of checking integrity should be used whenever possible. Multiple checks using different methods on a single input parameter adds to the overall quality or accuracy of the integrity assessment. What follows is a description of various methods for determining integrity. The discussion extends beyond sensor data integrity to address overall system health or system integrity as well as flight integrity.

Within the Resilient Autonomy Integrity Manager are four functionally different categories used towards monitoring system integrity:

1) Self-Health Monitors
2) Derived Data Functions
3) Dynamic Consistency Monitors
4) Outer-Loop Integrity Verification Monitors

Self-Health Monitors receive data directly from the sensor providing the data for the RTA. This category of monitor relies on that sensors self-assessment of data quality. Many time sensors have more data available internally to evaluate data integrity than what is made available external to the sensor.

Derived Data Functions combine multiple input parameters to compute a new parameter. These parameters are used by subsequent functions within the RTA architecture, both external to the Integrity Manager as well as internal to the Integrity Manager. An example of a derived parameter could be deriving an estimate of what aircraft angle of attack should be given sensor input from a normal accelerometer and airspeed.

Dynamic Consistency Monitors do sanity checks on input data. This could be range checking that airspeed is within normal operating limits of the aircraft, or that the derived angle of attack as mentioned in the previous paragraph is not close to matching sensed angle of attack. If this angle of attack, airspeed, normal acceleration cross check failed it indicates that one of those three parameters has failed and should not be used to drive a Recovery Control Function.

Outer-Loop Integrity Verification Monitors perform checks against a broader set of indirect parameters to identify abnormal flight behavior. For instance, a monitor that tracks throttle position or command and aircraft altitude. If maximum throttle is being commanded yet aircraft altitude has continued to decrease over some period of time, there must be something inherently wrong with the aircraft’s ability to maintain altitude, and if this persists, the aircraft will eventually crash. Some recovery function should be triggered to mitigate the eventual crash. Another example might be an RTA monitor persistence check. Such a monitor would track the number of times a safety monitor had been selected by the RTA switch. If say in the last 15 minutes there have been 3 engagements of the ground collision avoidance recovery function then there is obviously something inherently wrong with the Complex Function's command of altitude. A recovery function here might re-plan the mission at a higher altitude and decide to land as soon as practical. Outer-Loop Integrity Verifier Monitors identify basic conditions that indicate it is not safe to continue the mission as currently planned and some alternate course of action needs to be taken.

The Second RTA Switch for Re-Planning the Mission Due to Contingencies

The Integrity Manager function should have the following primary functions:

1) Managing the inputs to the RTA,
2) Determining the validity of that data used by the "RTA Monitors" and "Recovery Control Function,"
3) Determining the state of critical aircraft components through either direct sensor observation or through derived means. This is critical in order to determine whether an emergency system failure has occurred, and
4) Triggering what should be done if a failure were detected.

It could be argued that a function triggering an alternate behavior based on a failure within the RTA architecture should simply be another "RTA Monitor". This is perfectly true. However, RTA contingency functions reacting to emergency/failed conditions fall into a grouping of recovery control functions with their own unique requirements. For instance, deviations required when an emergency has been detected may require control solutions that persist for a substantial length of time, sometimes to the end of the flight. Therefore, it may be highly desirable or even necessary that the Monitors and

Recovery Control Functions of the more tactical safety systems (sense and avoid, ground collision avoidance, etc) continue to function and take momentary control to avoid breaching their boundaries while the strategic Emergency Recovery Controllers are in control of the aircraft. For this reason, the Resilient Autonomy team has mechanized a second RTA switch that is driven by the Integrity Monitors within the Integrity Manager. This second switch manages when and which Strategic Recovery Control Functions should control the flight of the aircraft and feeds into the original RTA Switch allowing the tactical Recovery Control Functions to remain available to control the aircraft if needed. Examples of Recovery Control Functions that may be engaged under these failed conditions are:

1) Rejoining the originally planned route
2) Modifying the planned route due to some event that has come up during the mission
3) Return to base
4) Land as soon as practical
5) Land as soon as possible
6) Safely ditching

In order to determine which of these Recovery functions is engaged in the event of a failure, the system must consider the consequence and probability associated with each action. The determination then becomes an optimization problem to reduce the total expected negative consequence. The famous “Miracle on the Hudson” mishap is an excellent example of this process as performed by a human pilot. During climb out from LaGuardia airport, US Airways Flight 1549 struck a flock of Canadian Geese and lost thrust from both engines. The pilot, Captain Sully Sullenberger, chose to ditch in the Hudson River and all aboard were saved. However, simulations of the mishap showed that if Sully had made an immediate decision, after the bird strike, to return to LaGuardia, his disabled aircraft very likely could have been able to make the nearest runway. However, in that instance, there was no way to be certain the aircraft could successfully make the runway back at LaGuardia. The runway was near the glideslope limits of the aircraft, and variations in that glideslope due to winds alone placed in question the certainty of making the runway. Added to this, something unexpected had just happened that had compromised the aircraft's airworthiness further adding to the uncertainty of the aircraft's real glideslope capability. Finally, it takes time to assess the situation (this is stressed within the ASTM Standard) and in that assessment time, the probability of making the runway was diminishing. These uncertainties combined to pose the real and growing possibility that the aircraft could end up not making the runway and if that outcome happened, the consequences were dire. Falling short of the runway meant coming down in the dense urban environment of The Bronx, NY, resulting in a complete loss of all on board as well as a large number of fatalities and damage on the ground. Alternatively, the Hudson River was clearly reachable. This option provided an outcome where there was a large available area for landing to accommodate possible variances in glideslope capability, very unlikely loss of life on the ground, and a lesser chance of loss of life of those onboard the aircraft. The Hudson was clearly the right choice. The factors driving that choice were not the possibility of a good outcome; but instead, the possibility of a bad outcome eliminating that option as an acceptable choice. The integrity manager must make a similar determination if the system can meet or exceed the capabilities of the human pilot. Whether or not the individual expected consequence calculations are accurate will be most important to determining the best course of action.

Calculation of the expected consequence of a specific mission plan could be relationship summing the consequences, C, and multiplying each by some portion of their respective probability, P, in the following manner:

$$E[C] = \sum_i C_i * P_i$$

This is a typical calculation of an expectation value.

Individual consequences and probabilities for each potential outcome associated with a Recovery Function could be found using the following example process. The first step would be to model the ideal path or trajectory that the Recovery Function will execute. During each section of the modeled maneuver, the aircraft will have a “Reserve Control Authority” which is a measure of the aircraft’s ability to overcome disturbances and remain on the ideal path. When this quantity is evaluated against potential disturbances, uncertainties in prediction from the ideal path can be calculated. This ideal path combined with the associated uncertainties produce a probabilistic distribution of potential trajectories. These are then compared against geographical data to determine if any intersection of trajectory with geography occurs and the probability of that intersection. (Here an intersection is usually the potential for a landing or crash.) If an intersection occurs, then the geographical data is queried at that location to determine the potential consequence (typically in the form of risk to the aircraft, ground populations, property, or the environment) that the intersection will produce. Combination or evaluation of all possible consequences in some way would then score the emergency function and produce a hierarchy which could then be used to decide control authority.

Splitting up the Trusted Controller

Experimental setups on two aircraft were used to test RTA concepts. Each aircraft already had an autopilot certified to a level commensurate with the aircraft’s certitude. To minimize the complexity, cost and level of effort of the integration, the team utilized the existing autopilots to perform a portion of the Recovery Control Function. The design called for capture commands to be sent to the autopilot from the RTA architecture. This very approach is described within the requirements of the Recovery Control Function within F3269.

The Monitors used by the team evaluate many recovery options and only request to engage the Recovery Control Function when the last option is about to breach the Monitor’s boundary. The benefit of this method has been documented in both the Automatic Air Collision Avoidance Evaluation of the F-16 (ref. 4) and the Flight Evaluations of a number of NASA Armstrong Ground Collision Avoidance Systems (ref. 5 & 6). Because the Monitor determines which option is the best for recovery, the Monitor must communicate the capture values of the desired maneuver option to the Vehicle Management System through the RTA Switch. Therefore, the Monitor fulfils a portion of the Recovery Control Function.

To ease the integration of the Resilient Autonomy RTA architecture with a variety of autopilots, the capture values from the RTA switch are passed through a software isolated Command Coupler before going to the autopilot. This Command Coupler converts the maneuver request into the format, units and axes used by that specific autopilot. In this fashion, software changes to adapt the system to a different autopilot interface are isolated to only the Command Coupler.

Conclusion

The Resilient Autonomy team has adapted EVAA to five different aircraft over the past five years. These aircraft had highly different Vehicle Management Systems, flight characteristics and concepts of operation. They ranged from a quad-copter, to transition UAVs, to general aviation aircraft with a pilot in the loop. The effort of adapting EVAA was considerably less than seen with previous systems. The above thoughts and techniques are the byproducts of this work. In summary, increased design flexibility and system robustness come from not just one RTA architecture, but from a network of RTAs. These results are captured in the revised RTA diagram presented in Figure 2. Future planned work includes flight testing of the EVAA RTA network, which will begin around the end of 2019 and conclude in early 2021 on a 115 pound vertical takeoff, transition UAV. Other future work, which will be necessary to the inclusion of these concepts, will certainly involve analyzing the interactions of separate RTA monitors and switches on the behavior of the system as a whole. In addition, formalizing the analysis of emergent behavior seen with multiple simple and temporary controllers on safety, an effect that has been dubbed "computational agility."

References

[1] J. D. Schierman *et al.*, "Runtime assurance framework development for highly adaptive flight control systems," Barron Associates, Inc. Charlottesville, 2015.

[2] M. Clark *et al.*, "A study on run time assurance for complex cyber physical systems," AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH AEROSPACE SYSTEMS DIR, 2013.

[3] L. R. Hook, M. Clark, D. Sizoo, M. A. Skoog, and J. Brady, "Certification strategies using run-time safety assurance for part 23 autopilot systems," in *2016 IEEE Aerospace Conference*, 2016, pp. 1–10.

[4] D. E. Swihart *et al.*, "Automatic ground collision avoidance system design, integration, & flight test," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 26, no. 5, pp. 4–11, 2011.

[5] M. A. Skoog, K. Prosser, and L. Hook, "Ground collision avoidance system (iGCAS)," 9,633,567, 25-Apr-2017.

[6] ASTM, "ASTM 3269-17 Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions," 2017.

[7] S. P. Cook, "An ASTM Standard for Bounding Behavior of Adaptive Algorithms for Unmanned Aircraft Operations (Invited)," 2017.

[8] L. R. Hook, M. Skoog, M. Garland, W. Ryan, D. Sizoo, and J. VanHoudt, "Initial considerations of a multi-layered run time assurance approach to enable unpiloted aircraft," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–11.

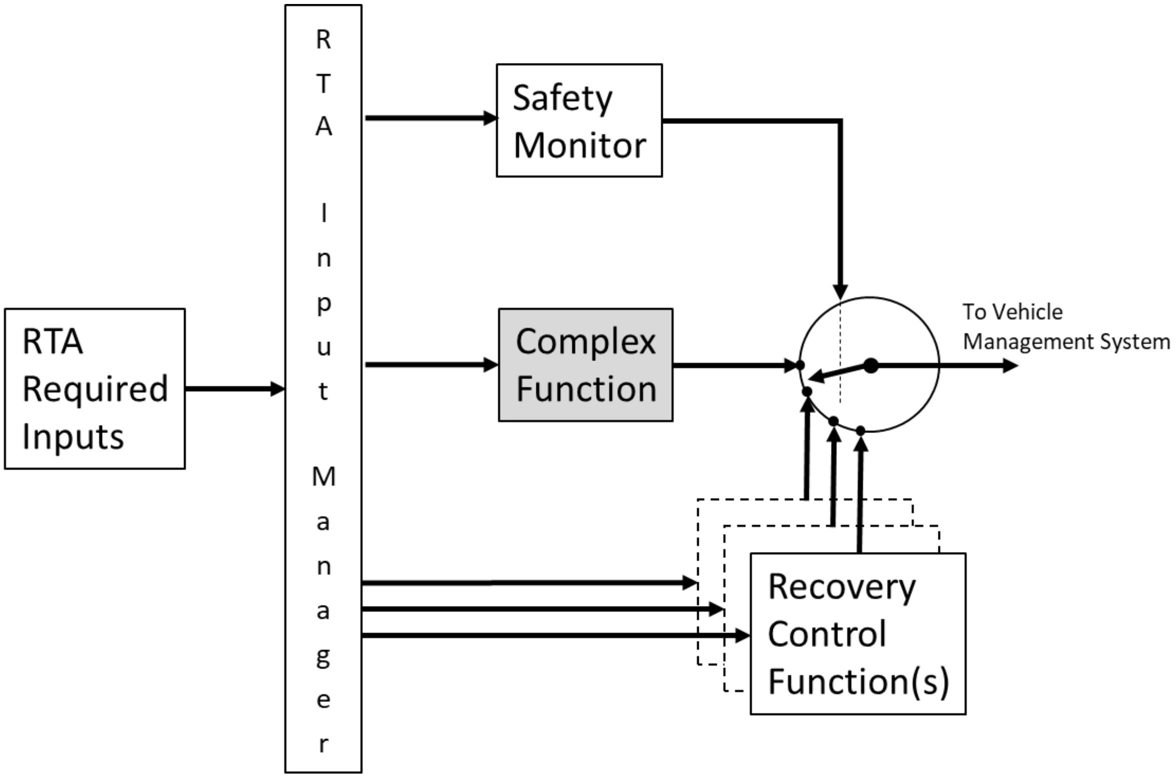

Figure 1. ASTM F3269 RTA Architecture

EVAA RTA Network Architecture

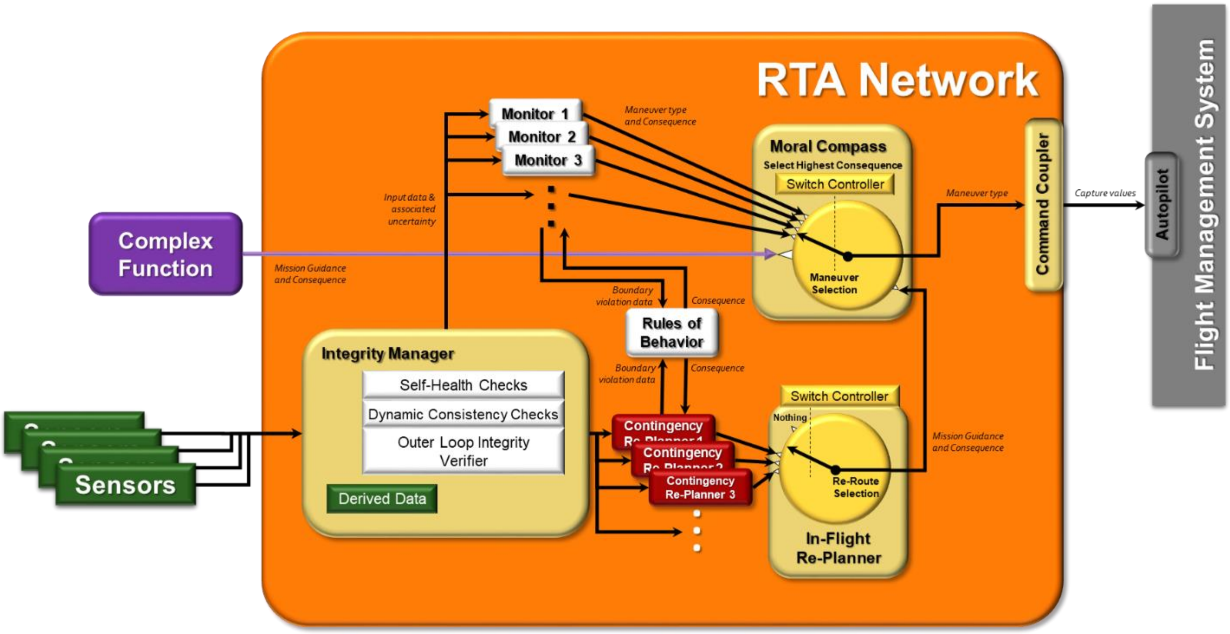

Figure 2. The Simplified EVAA RTA Network