Development of an Autonomous Distributed Fault Management Architecture for the VISORS Mission

Antoine Paletta[1] and E. Glenn Lightsey[2]
Georgia Institute of Technology, Atlanta, GA, 30318

CubeSat formations have been identified as a new paradigm for addressing important science questions, but are often early adopters of new technologies which carry additional risks. When these missions involve proximity operations, novel fault management architectures are needed to handle these risks. Building on established methods, this paper presents one such architecture that involves a passively safe relative orbit design, interchangeable chief-deputy roles, a formation level fault diagnosis scheme, and an autonomous multi-agent fault handling approach. This architecture focuses on detecting faults occurring on any member of a spacecraft formation in real time and performing autonomous decision making to resolve them and keep the formation safe from an inter-satellite collision. The NSF-sponsored Virtual Super-resolution Optics with Reconfigurable Swarms (VISORS) mission, which consists of two 6U CubeSats flying in formation 40 meters apart as a distributed telescope to study the solar corona, is used as a case study for the application of this architecture. The underlying fault analysis, formulation of key elements of the fault detection and response strategy, and the implementation as flight software for VISORS are discussed in the paper.

I. Introduction

Background on Fault Management for Space Missions

Fault management (FM) is defined as a set of strategies, design decisions, and requirements that help ensure mission success and mitigate risks during operations. Typically, this involves a systems engineering effort during the design phase to build redundancy into a spacecraft, a hardware and software engineering effort to allow the flight system to detect and respond to faults, and an operational engineering effort to enable the ground segment to react to issues on orbit and restore the mission to nominal operations. This paper will focus on the latter two of the three efforts listed above – i.e. devising the reactive strategies and their software implementations to manage faults occurring in a spacecraft formation on orbit.

Typically, for legacy spacecraft missions, these reactive strategies involve diagnosing faults as they occur in real time, attempting to isolate the subsystem involved, and implementing corrective actions such as switching to backup systems or putting the spacecraft into a "safehold" mode. Whenever the next ground contact occurs, operators will receive an alert that the spacecraft has entered a safe mode and carefully examine telemetry to understand the root cause of the issue. Once the fault is isolated, recovery strategies are implemented to mitigate its effects. This can involve reconfiguring the spacecraft, power cycling subsystems, or patching flight code to restore the spacecraft to nominal operations. The effectiveness of these strategies is dependent on the spacecraft's fault tolerance capabilities and the mission's operational constraints.[1]

[1] Graduate Research Assistant, Daniel Guggenheim School of Aerospace Engineering, apaletta3@gatech.edu.
[2] Professor, Daniel Guggenheim School of Aerospace Engineering, glenn.lightsey@gatech.edu

Background on Spacecraft Formation Flying and Proximity Operations

As space missions become increasingly complex, the need for more affordable ways to achieve such mission objectives grows. One potential way of reducing cost for more complex missions is through distributed spacecraft systems, which are a system that consists of multiple spacecraft, which work together to achieve the mission objectives that would otherwise be significantly more expensive or infeasible to perform with a single spacecraft.[2,3,4,5] A subcategory of DSSs relevant to this paper is formations or swarms, where the position of multiple spacecraft is controlled relative to one another. More specifically, precision formation flying (PFF) occurs when the relative position of multiple vehicles must be controlled autonomously on-board in a continuous manner with a high level of accuracy, due to the stringent relative state requirements. Usually, PFFs involve some sort of proximity operations for a portion or the entire duration of the mission.

While the PFF concept allows otherwise inaccessible science to be conducted, it also presents unique challenges. Due to the close proximity of multiple spacecraft to each other and the effect of unmodelled perturbations on the relative orbit, there is usually a risk of an inter-satellite collision. This collision could occur if the formation is incorrectly actuated and one spacecraft performs a maneuver that directly leads to a collision, or if an individual spacecraft loses the ability to maneuver and passively drifts into another one. In many cases, the time between loss of control of one spacecraft to an inter-satellite collision is shorter than the time it would take for the ground to be alerted and plan an appropriate response. Therefore, fault management designed to mitigate this sort of intersatellite collision risk must be designed with autonomy in mind and be able to react to anomalous scenarios on orbit without the ground in the loop.

Within the space industry, there seems to be a lack of a consistent and unified approach to fault management, with many legacy space missions describing their approach as more of an art form than a science. In light of this, this paper aims to develop a unified approach to the design, development, and implementation of a fault management architecture for spacecraft formation flying missions involving proximity operations. Moreover, generalizable guidelines and methodologies will be discussed, and the VISORS formation flying mission will be used throughout the paper as a case study in how to implement this fault management architecture.[6,7]

II. VISORS Mission Overview

VISORS is a National Science Foundation (NSF) space physics mission which will detect and study fundamental energy-release regions in the solar corona. The VISORS mission will image extreme ultraviolet (EUV) features on the Sun at a resolution of at least 0.2 arcseconds from Low Earth Orbit (LEO). To accomplish this objective, VISORS will use a pair of formation flying 6U CubeSats: one of which carries the observatory optics while the other contains the detector instrument. They will line up once per orbit at a distance of 40m to form a distributed space telescope and capture an image, as shown in Figure 1. VISORS will demonstrate several technologies key to precise formation flying including intersatellite links, autonomous relative maneuver planning and control, and miniaturized on-board propulsion.

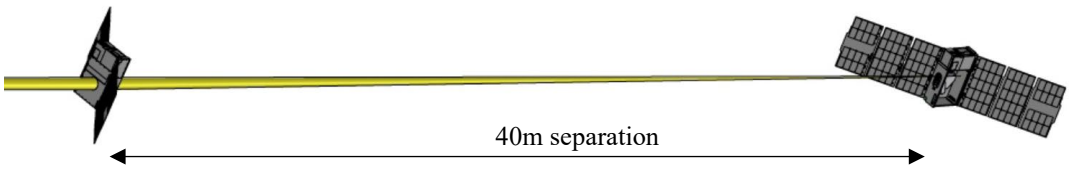

Figure 1. VISORS formation during alignment.

VISORS GNC Requirements and Relative Orbit Design

In order to capture this image of the solar corona, the formation will have to meet some very stringent relative position and velocity requirements at the moment of observation to ensure that the image is both in-focus and free of blur. These requirements are shown in Figure 2 and include a longitudinal separation requirement between the optic and detector of 40m ± 15 mm, a lateral alignment requirement of ± 17.5 mm, and a relative velocity requirement of less than 0.2 mm/s. These three requirements need to be satisfied during the 10 second observation window as the OSC drifts in front of the DSC that takes multiple images.

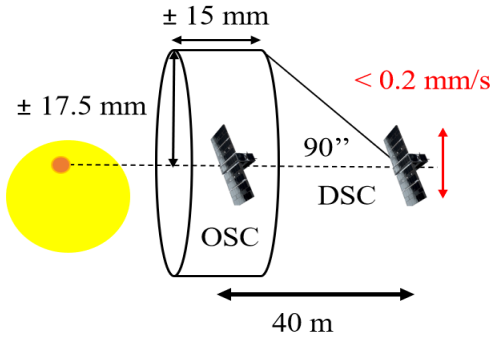

Figure 2. Relative position/velocity requirements imposed on formation during observations.

In addition to meeting these inertial pointing and stability requirements, the relative orbit is designed to address the risk of an inter-satellite collision and has safety baked into its design by utilizing two techniques:

1. Passive safety: a duration of time during which there is a guarantee that no inter-satellite collision will occur, even under the effect of worst-case orbital perturbations.
2. Active safety: in the event that an imminent inter-satellite collision is detected, a collision avoidance maneuver (CAM) designed to increase the inter-satellite separation can be performed.

This two-pronged approach to orbital safety allows the team to significantly reduce the risk of a debris-causing collision on orbit, and deciding how to leverage the passive and active safety elements with on-board fault management will be a focus of the rest of this paper. Since the along-track separation between both spacecraft is subject to the high uncertainty under the effects of differential drag and orbital perturbations, a sufficient passive safety margin is best achieved using a relative orbit design called “eccentricity/inclination vector separation”.[8] This method, resulting in a relative orbit shown in by the blue line in Figure 3, allows a minimum separation to be maintained at all times in the plane perpendicular to the along-track direction. For VISORS, when both spacecraft are in their closest configuration, the passive safety guarantee is ~2 orbits (~3 hours). The active safety CAM can be performed autonomously, and aims to rapidly increase the inter-satellite separation, thus creating a much larger margin of passive safety and enough time for the ground to get involved and address any anomalies. An example delta V for such a maneuver is shown in Figure 3 as the pink vector and the resulting trajectory is shown in red; the maneuver simultaneously increases the RN separation while also introducing an along-track drift.[9]

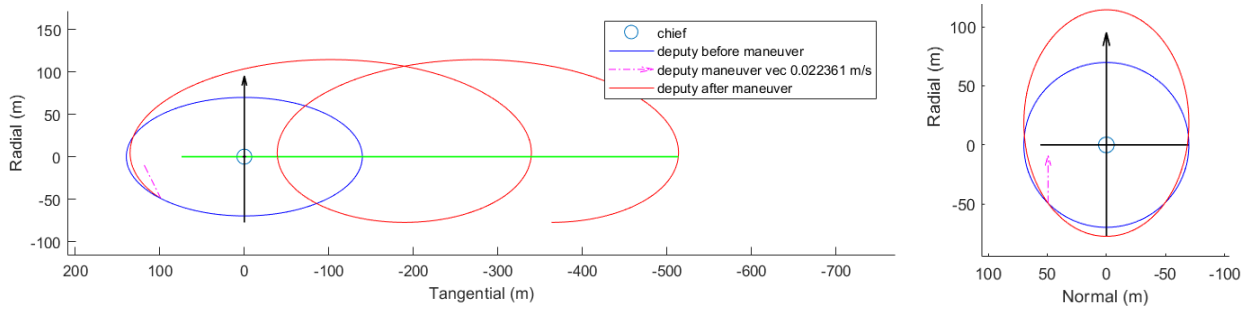

Figure 3. Illustration of relative motion before (solid line) and after (dashed line) a collision avoidance maneuver in the RTN frame.

The final important aspect of the relative orbit design is how the formation is controlled. A chief-deputy architecture is utilized, with only one spacecraft allowed to maneuver at any given time. The chief does not maneuver and sits at the center of the Clohessy-Wiltshire (CW) non-inertial frame, while the deputy does maneuver and attempts to control its relative orbital element (ROEs) with respect to the chief.[10] For the rest of this paper, the chief will be referred to as the "passive" spacecraft since it does not maneuver, while the deputy will be referred to as the "active" spacecraft since it does maneuver. Crucially, these formation roles are designed to be interchangeable, so either the OSC or DSC can become the active spacecraft and control the relative configuration of the formation. This means that from a formation control perspective, both spacecraft are identical and are capable of performing maneuvers, allowing for the balancing of propellant consumption between the two as well as extra redundancy if there are propulsion failures on one spacecraft. However, it is important to note that the relative orbit is not designed for both spacecraft to maneuver at the same time, and this distinction will come into play in a later discussion about role switching.

VISORS Payload and Concept of Operations

The VISORS mission will fly in a Low Earth Orbit (LEO) sun-synchronous orbit and make use of several subsystems on board to enable the aforementioned autonomous navigation and control to the required levels of accuracy. The DSC and OSC are both built using the commercial-off-the-shelf (COTS) 6U BCT XB1 spacecraft bus. It provides the spacecraft's command and data handling (C&DH), power generation and storage, attitude determination and control (ADCS), space-to-ground communications, and an L1/L2 GNSS antenna. Adding to this, the VISORS team is integrating a payload consisting of a 3D-printed cold gas propulsion system, with 6 orthogonal nozzles to provide delta V in any direction without any attitude constraints, an intersatellite-link capability with 6 patch antennas (one on each face of the spacecraft for full sky coverage), science instruments for each respective spacecraft to make coronal observation, and a hosted software application (HSA) that lives on a partition of the BCT Xilinx flight computer.[11]

The HSA contains all the mission-specific operational software, such as GNC algorithms that utilize differential carrier phase GNSS measurements exchanged between both spacecraft over the ISL to achieve relative position estimates down to the millimeter level, and decision making/fault management logic that controls the different payload subsystems and interacts with the other spacecraft. An important consideration for the VISORS fault management approach is that the HSA with all its mission specific logic responsible for controlling the formation exists in the "hosted software payload" paradigm where it is only alive and running when the BCT bus is in a nominal operating mode. As can be seen from Figure 5, this means that the HSA and thus formation control are only on in BCT's "Fine Reference Point (FRP)" mission mode, and is turned off in any of BCT's safe or manual modes ("Launch, "Sun Point", and "Survival"). Additionally, the mode transitions visible in the diagram illustrate that there are only autonomous transitions into BCT's safe modes, but none out of its safe modes. Therefore, the bus's own fault management system detects a bus-level fault and transitions to its safe mode, the HSA will be turned off on that spacecraft until the ground can establish contact, resolve the issue, and command a transition back into FRP.

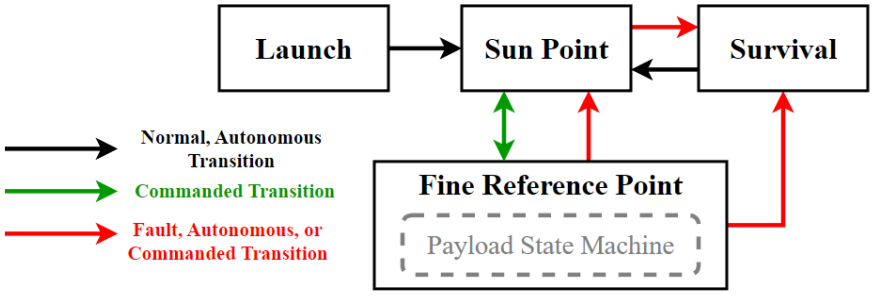

Figure 4. BCT bus mission modes.[12]

III. Fault Analysis

In order to formulate a fault management architecture, one must start by analyzing the risks and failures that could occur during the mission, and then determine which ones can be mitigated and at what cost. A comprehensive fault analysis was performed, including a mission risk categorization and a Failure Modes, Effects, and Criticality Analysis (FMECA).[13] A distributed system such as VISORS is unique in terms of fault analysis compared to a single spacecraft mission as there is not just one but two different types of total system failures possible. There are premature end of mission (EOM) system failure events that would likely result from a permanent combination of faults in the subsystems on either spacecraft, and then there are inter-satellite collision system failure events that would likely result from a temporary combination of faults in the subsystems involved in formation control on either spacecraft. In terms of risk mitigation, the permanent subsystem fault EOM system failures are addressed by having redundancy in certain subsystems (such as the propulsion system) that is a direct result of having two spacecraft. However, the inter-satellite collision EOM system failure is best addressed with operational mitigation strategies and functionality built into the software and orbit design. Therefore, the fault analysis and the rest of the paper will focus on addressing the inter-satellite collision EOM system failure.

Table 1. Formation Flying Functionalities Description

FFF	Description
FSW Application	The FSW application (the HSA for VISORS) needs to be on and running in order to make decisions and have the capability to manage faults.
Inter-satellite Link	The ISL needs to be working in order for multiple spacecraft to coordinate with the other spacecraft and make formation level decisions.
Relative Navigation and Maneuver Planning	The relative navigation filter needs to be converged in order to have a precise state estimate and be capable of predicting the relative motion of the rest of the formation.
Maneuver Execution Ability	At least one spacecraft needs to be capable of performing maneuvers to control the formation's configuration and separation between individual spacecraft.
All of these FFFs need to be functioning on each spacecraft in order for the formation to maintain active control. If one or all of these cease to function, then the formation is no longer actively controlled which poses a potential collision risk since the formation is only passively safe for a short buffer duration.	

In order to safely control the formation with both spacecraft in such close proximity, certain subsystems/functionalities must be continuously operating nominally, these are shown in the Table 1 above. A temporary fault or combination of faults in any of these FFFs can lead to a degradation of the formation's passive safety over time, potentially leading to an inter-satellite collision. The severity of the collision risk posed is related to the fault's duration, the specific combination of faults occurring across either or both spacecraft, and the extent to which each spacecraft's FFFs are affected.

Risk Categorization

In order to categorize the different risks and perform an effective fault analysis, each relevant combination of FFF faults is assigned to an inter-satellite collision failure scenario, which is then ranked using the Risk Priority Number (RPN) method. Each scenario has three defining metrics: severity, likelihood, and observability, which are rated on a scale of 1-5 and multiplied together to obtain an RPN rated on a scale of 1-125. The severity metric rates the level of collision risk for each failure scenario's

outcomes; for example, scenarios where formation roles can be switched and a CAM can easily be performed to increase spacecraft separation result in a low collision risk. The likelihood metric rates the probability of individual FFF faults occurring during the mission's lifetime and uses NASA Goddard's standard FMECA probability categorization scheme for the 1-5 rating.[14] The overall likelihood rating assigned to each failure scenario corresponds to the probability product of the individual faults making up that scenario. Finally, the observability metric rates the difficulty the FSW App experiences when attempting to properly diagnose and respond to faults occurring in each failure scenario given its limited perspective – i.e. if the ISL on one spacecraft stops working, the FSW App loses visibility over the state of the other spacecraft.

FMECA Matrix and Analysis

The FMECA for the inter-satellite collision EOM system failure is shown below in Figure 5. When performing this analysis, it is important to examine fault detection and response from the perspective of the logic running in a spacecraft's FSW App, and the limitations in full and perfect knowledge of the rest of the formation's statuses. This means that faults occurring on a spacecraft in question (referred to as the "local") must be considered as well as faults occurring on a spacecraft other than the one in question (referred to as the "remote"). Since the VISORS formation has two spacecraft, an FFF fault combination happening on the first, second, or both spacecraft are considered for the FMECA shown below. More complex combinations of these FFF faults are not considered here due to the fact that the presence of a fault in one FFF usually cascades down into other FFFs, causing them not to work either. For example, if the FSW App is not running, none of the other FFFs can function, or if navigation and maneuver planning is not working, then it doesn't matter whether or not you can execute maneuvers.

Scenario #	Formation Flying Functionalities								Severity	Likelihood	Observability	RPN
	Active				Passive							
	Hosted Software Application	Inter-sat Link	Navigation & Maneuver Planning	Maneuver Execution	Hosted Software Application	Inter-sat Link	Navigation & Maneuver Planning	Maneuver Execution				
1	■								4	2	4	32
2					■				3	2	4	24
3	■				■				5	1	5	25
4		■							4	4	4	64
5						■			4	4	4	64
6		■							4	2	4	32
7			■						3	2	2	12
8							■		1	2	2	4
9			■						5	1	3	15
10				■					3	4	1	12
11								■	1	4	1	4
12				■					5	1	1	5
13	■	■	■	■					4	3	4	48
14					■	■	■	■	3	3	4	36
15	■	■	■	■	■	■	■	■	5	1	5	25

Figure 5. FMECA Matrix

As can be seen from the FMECA matrix for VISORS, the highest RPNs indicate which are the most important scenarios to mitigate when designing the fault management architecture to handle. Highlighting some examples, scenarios #4 and #5 are concerning as they would result in a relatively high collision risk, are fairly likely to occur at least once over the mission lifetime (due to the ISL subsystem not having flight heritage), and the HSA would have difficulty properly diagnosing where the fault is occurring since neither spacecraft would be aware of the other's status without an ISL. Similarly, scenario #13, which represents the effect of a sudden Bus safe mode (all the payload is turned off) on the local active spacecraft is concerning because it would no longer be able to maneuver, it is fairly likely to occur due to something like

a Single Event Upset (SEU) over the mission lifetime, and it would be difficult to diagnose from the remote spacecraft because it would suddenly stop hearing from the local with no warning.

Based on the analysis carried out above, the FM architecture should be able to handle (wherever possible) scenarios where there is an unexpected bus safe mode/reset, an ISL failure, or issue with maneuver planning/execution on either spacecraft. Additionally, the FM architecture should be able to handle (whenever possible) certain common issues that don't necessarily affect the FFFs but could damage the hardware on either spacecraft (similar to single spacecraft FM).

IV. Multi-Agent Fault Handling Strategy

The multi-agent fault handling strategy discussed below is an elegant way of exploiting the over-actuated nature of a spacecraft formation where all members have maneuvering capability. This redundancy helps eliminate many single points of failure for spacecraft formations and allows the mission to continue even in the presence of multiple failures. Since all agents in the formation have the same capabilities with respect to formation control, they can cooperate to detect a fault and select the "healthiest" or "best suited" spacecraft(s) to take on extra maneuvering responsibility and make up for an underperforming/failing agent(s). There are two potential outcomes of successful fault handling, either maneuvering responsibilities are redistributed throughout the formation, thus returning it to "full active control"; or action is taken to sufficiently separate the members experiencing maneuvering faults from the rest of the formation such that they no longer pose a collision risk. The goal of this strategy is to act as an autonomous safety net for a formation flying mission by detecting faults in one/multiple members of a formation and choosing a response to safe the formation from an inter-satellite collision risk before the ground has a chance to intervene. Then, once the formation is safe, the ground can take their time to determine the root cause of the issue and return the formation back to nominal operations. Even though this multi-agent fault handling strategy was designed for the VISORS mission, it can be applied to formation missions involving proximity operations with any number of spacecraft; as such the discussion below will discuss the key elements of the strategy in a manner more generally applicable.

Distributed Characterization of Formation Health

In order to apply this multi-agent strategy, the state of health (SOH) of each member of the formation needs to be shared between all other members. In this context, the SOH of each spacecraft will be defined according to a diagnosis comprised of macro-level statuses that correspond to the FFFs. These are as follows:

1. **FSW Application** (Running or not running)
2. **ISL connected** to other members of the formation (Connected or Not for each member)
3. **Navigation & Guidance** Working or Not Working
4. **Control & Propulsion** Working or Not Working

If all four of these "formation statuses" are diagnosed as "working" or "true" on the local spacecraft, then it is considered perfectly healthy and can contribute to active formation control (if it is currently active it is already doing so and if it is currently passive it could be called upon to replace another malfunctioning member). If some or all of the formation statuses on the local are diagnosed as "not working" or "false", then its ability to contribute to active maintenance of the formation is hampered, depending on which statuses are false.

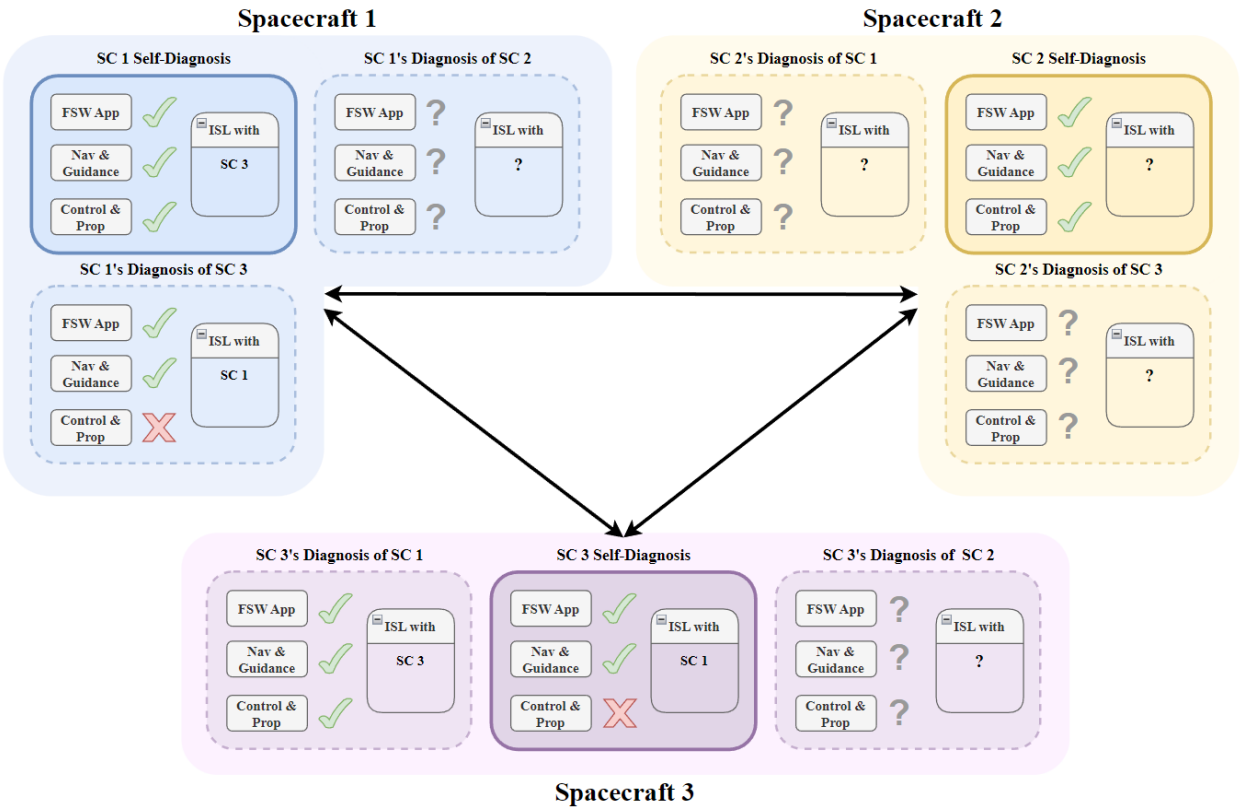

Figure 6. Formation statuses sharing scheme across a formation of three spacecraft.

An example of this distributed formation health characterization applied to a sample formation of three spacecraft is shown in Figure 6. Each spacecraft performs a self-diagnosis of its formation statuses based on telemetry coming from all local subsystems and sends this diagnosis out to all remote spacecraft (ideally). Simultaneously, the local spacecraft is receiving diagnoses of all (ideally) the remote spacecrafts' formation statuses and storing the most recent version of these locally. It is up to the specific implementation of this SOH sharing scheme if one would prefer to send all the raw telemetry values for each spacecraft to perform its own diagnosis of the rest of the formation instead of directly sending the "processed" self-diagnosed statuses. This will likely depend on the ISL's data capacity and amount of raw telemetry needing to be shared.

Figure 6 illustrates a range of cases, where spacecraft 1 is considered perfectly healthy, spacecraft 2's ISL is not connected with any other spacecraft, and spacecraft 3's control & propulsion is not working according to their own self diagnoses. All three spacecraft attempt to broadcast their self-diagnoses to the others, but only spacecraft 1 and 3 are able to effectively exchange theirs. This results in both of these spacecraft having full awareness of each other's diagnoses, which are stored locally on each spacecraft. However, due to spacecraft 2's malfunctioning ISL it is not able to connect to spacecraft 1 or 3, and so is only able to store an "invalid" diagnosis of spacecraft 1 and 3. Similarly, spacecraft 1 and 3 store an "invalid" diagnosis of spacecraft 2. Temporary outages in inter-satellite communications is almost certain to occur on orbit due to the lack of flight heritage of this technology, and thus it is critical to ensure that this scenario is taken into account in this health status sharing scheme.

Strategy for Responding to Faults

Designing logic that runs on each formation member and acts on this distributed knowledge of the formation's SOH in a coordinated manner necessitates adhering to certain assumptions/principles. If any of these assumptions are invalidated, the formation status sharing scheme can no longer be considered valid

and it is not advisable for a response to be executed autonomously. This distributed fault response approach has many advantages, but it is important that it be implemented in such a way that a deterministic outcome can be predicted for any possible contingency scenario. There are multiple examples of rigid autonomous logic onboard proximity operations mission in the past behaving in unpredictable ways because logic was autonomously executed even though its underpinning assumptions had been invalidated.[15]

Table 2. Categorization of and rationale for multi-agent fault handling principles.

Principle	Rationale
1. Each member will attempt to determine how "healthy" and capable of executing a formation level response it is compared to the other formation members.	Each member of the formation must continuously be evaluating its health relative to other members so that if an anomaly occurs with one member, it is straightforward to decide which other member is best suited to help take over the formation control responsibilities of the failing member.
2. If a formation member is not receiving formation health information, it cannot execute any formation level responses.	If the local spacecraft doesn't know the statuses of the other formation members, it is better not to act than to act on incorrect information.
3. If a formation member is receiving up-to-date formation health information, and it is healthy relative to the other formation members, it will assume responsibility execute any formation level responses.	Given that disconnected members will not perform formation level responses, connected members must assume this responsibility to maximize the number of scenarios in which an FFF fault can be resolved.
4. If two spacecraft are temporarily not connected with an ISL, both must assume responsibility for the ISL blackout.	Due to the way links work, there is no reliable way to tell if the link issue is on the local end or the remote end. The only way to tell if a link is connected is if you are continuously received data over it. Therefore, both spacecraft should assume they are responsible for the link blackout and act more conservatively.
5. Only a currently active spacecraft can delegate its role to a currently passive spacecraft, not vice versa.	This mechanism prevents too many spacecraft maneuvering at one time when they are not all supposed to. It ensures that if a link failure occurs during a role switch on VISORS for example, both spacecraft cannot become active. The active spacecraft must demote its role to passive before promoting the remote spacecraft to active.

The algorithm shown in Figure 7 illustrates a way to choose a response given an anomalous diagnosis of formation statuses from the local and a remote. The intention is that it would be run on board each spacecraft in the formation as they simultaneously diagnose and share that a fault has occurred on one spacecraft. In this case, the anomalous spacecraft would allow any other spacecraft running this algorithm to switch roles with it or take over formation control responsibilities. The algorithm shown in Figure 7 only demonstrates how to apply the multi-agent fault handling strategy across a formation of two spacecraft like VISORS, but it could be adapted to handle more spacecraft.

The algorithm in Figure 7 shows a linear progression through the formation statuses for the local and remote, as each status' value impacts whether the next one in line needs to even be considered. There are four possible "outcomes" of a fault response: a CAM is performed to rapidly increase spacecraft separation, a slow "transfer out" to a larger holding relative orbit to slowly begin to increase spacecraft separation, continuing science operations, and ceasing formation control (local spacecraft stop maneuvering) to wait for the ground to handle the issue. Additionally, there are also three role responses possible: A currently active spacecraft sending a role switch command to the passive spacecraft to delegate its active role to it, a

currently passive spacecraft receiving that role switch command to promote itself to the active role, and no role switching.

The diagram is entered via the blue dot, and the first consideration is to determine if there is an imminent collision risk that has been detected and perform a CAM if it has. The CAM "outcome" is considered before all the others as it is the most time sensitive one. The next three statuses, the FSW App Local, ISL, and FSW App Remote represent a situation where both spacecraft may have invalid statuses of each other's health, and so it is best to act conservatively. If the local's FSW App asserts at any point during the mission, an emergency role switch command to be sent over the ISL to the remote can be built into the assert handler to alert the remote spacecraft that the local is about to drop offline. In this situation, from the perspective of the remote, it can receive this command and switch roles accordingly. However, if no messages are received indicating a FSW assert, then the ISL (on both according to principle #4) is assumed to be the source of the communications blackout, and in this case no role switch should be performed. In these three situations, there is a common "outcome" where the currently active spacecraft should cautiously start increasing its inter-satellite separation based on its last known state of the remote. If all three of these statuses are nominal, then the next group of four statuses (Nav & Guidance Local/Remote and Control & Prop Local/Remote) can be considered with the assumption that the local and remote spacecraft both have perfect knowledge of each other's statuses. The decision logic here is rather simple, if the local spacecraft has a capability that the remote doesn't, then role should be switched to allow the more capable spacecraft to take on the responsibility of formation control, and vice versa. In this case, the "outcome" is that formation proximity operations can resume as normal. If neither spacecraft have a certain FFF capability, then there is no role switching needed. The "outcome" here is that both spacecraft stop maneuvering and wait for the ground to intervene. If both spacecraft have a FFF capability, then again no role switching should occur.

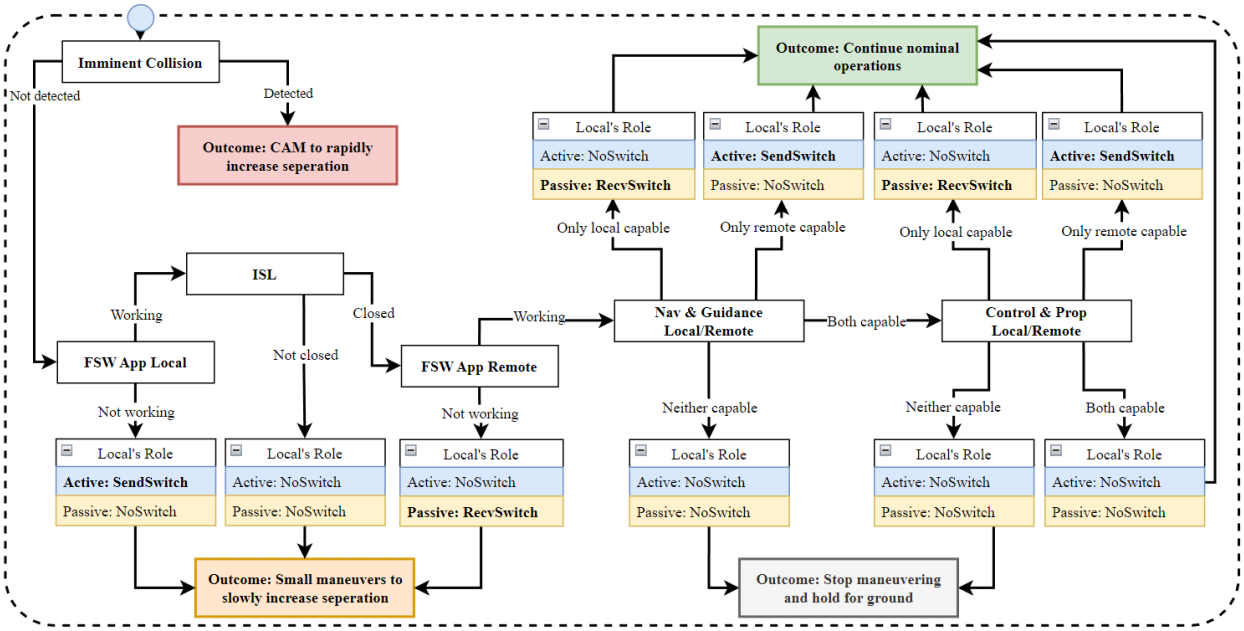

Figure 7. Decision algorithm to run on each spacecraft and illustrate how to resolve faults.

In practice, this algorithm would be called upon to choose a response on both spacecraft simultaneously, with each spacecraft having a mirrored version of each other's formation statuses. In this manner, both spacecraft will end up choosing a congruent role response (either no switch or a send-receive switch pair) and the same "outcome" response. This algorithm only addresses the problem of selecting one response in a coordinated manner across two spacecraft, but it could be extended to a larger formation to assign different responses to different spacecraft to resolve multiple faults occurring simultaneously.

V. VISORS Contingency Operations and Mission Modes

The VISORS mission makes use of a standard set of nominal operations modes to execute its concept of operations, as well as two non-traditional safehold modes to be entered into in the event of an anomaly. In addition to these specific mission modes, the higher-level concept of an "operating regime" is considered to develop a set of operating rules that allow the members of the formation to safely execute autonomous fault responses using the algorithm discussed above.

Nominal and Off-Nominal Mission Modes

As previously discussed, once the BCT bus transitions into its FRP state and the hosted application is turned on, there exists multiple mission modes to operate the payload. Nominal mission operations for VISORS require maneuvering between a larger (~200m separation) standby orbit where housekeeping tasks are performed and a smaller (~40m separation) actively maintained science orbit where the observations are conducted. This concept of operations reflects these two major orbital configurations with a Standby and Science mission mode, as well as a Transfer mission mode for the transition between the two configurations. In nominal mission operations, the spacecraft will move from Standby mode to Transfer mode to Science mode, and back, multiple times over the lifetime of the mission.[12] Off-nominal operations can be classified into three main types, two of which are relevant to this paper. There is a preliminary operations mode, which will be used to commission the payload subsystems only at the beginning of the mission, a Safe mode, which will be entered in the event of an anomaly, and an Escape mode, which will be entered if a CAM needs to be performed. It should be noted that after the commissioning phase, the preliminary operations mode will be disabled, and the default mode that each spacecraft will boot into when the HSA turns on will be Safe mode. These modes and the transitions between them are shown in Figure 8.

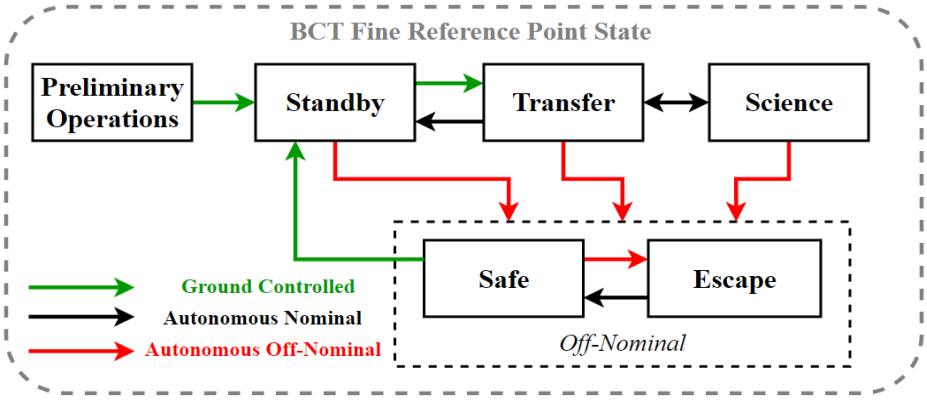

Figure 8. Payload defined nominal and off-nominal mission modes.

The off-nominal modes for VISORS have been designed as a combination of the Safe and Escape mode. First, the safe mode is non-traditional safehold move, compared to conventional spacecraft, where not all the subsystems can be turned off since the majority of them are necessary to execute the FFFs. Therefore, in the Safe mode, only the science subsystem is turned off to conserve power while the rest are kept on. The mode is designed to either be exited if a collision risk is detected and a CAM maneuver needs to be executed – in which case a transition to Escape mode occurs, or once the ground regains control of the formation it can manually command the spacecraft to exit Safe mode. Escape mode, on the other hand, can only be entered once autonomously from Safe or from any of the nominal mission modes, and is used to perform the CAM. Once the CAM has been executed, the spacecraft will transition back to Safe mode, with future autonomous transitions to Escape mode prevented, following the logic that a second CAM may not improve a situation if a first CAM had already been executed. While the two spacecraft will be in the same mode at any given time during nominal operations, it is possible for the spacecraft to not be in the same mode when dealing with contingency scenarios. For example, one spacecraft may enter Safe mode while the other remains in another mode.

Operating Regimes

The possibility of having multiple spacecraft in different modes at different times, as well as the need to have the autonomous formation behave in a consistent and predictable manner at all times creates additional challenges that can be addressed with the concept of "operating regimes". For the purposes of the paper, this concept is defined as a set of rules – either enforced through FSW or a mission operations team – that govern how the formation should behave autonomously and be interacted with from the ground under certain circumstances.[16] There are three different operating regimes defined for VISORS (shown in Figure 9) that capture the main aspects of autonomous formation operations: a ground controlled regime where both spacecraft are operated manually, an autonomous nominal regime where the formation is controlling itself while executing the mission, and an autonomous off-nominal regime where an anomaly has occurred and the formation is handling it autonomously before the ground is able to get involved.

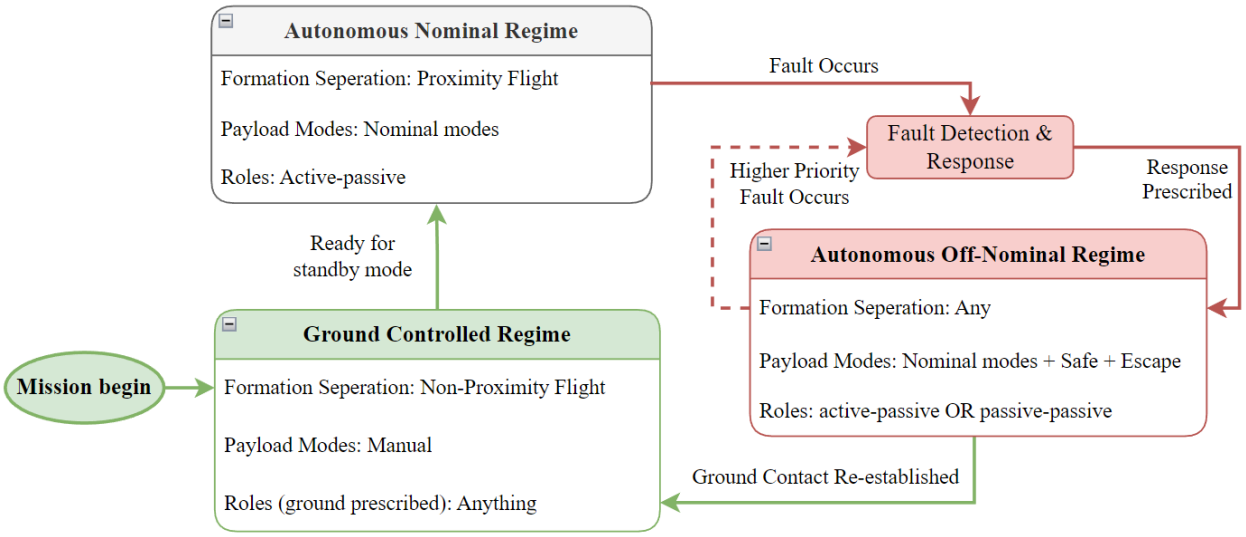

Figure 9. Three different operating regimes.

The mission will begin in the ground controlled regime, where both spacecraft will be commissioned and the ground will have full control over their configurations. This means that the roles and mission modes, on each spacecraft will only be set by the ground, and autonomous transitions between them are prohibited. This regime exists to allow the ground full manual control of the formation to perform system checkouts, on-orbit calibration, or functionality testing before/during the mission's science operations without having to worry about accidentally activating the autonomous FM safeguards. However, this operating regime is only allowed when both spacecraft have a large enough inter-satellite separation that it is safe to operate without the autonomous safeguards in place. For example, it would be perfectly safe to set both roles to active in order to execute ground planned maneuvers on both spacecraft to help calibrate their propulsion systems or help phase them together in preparation for the proximity formation acquisition.

Once both spacecraft are declared fully operational and are in a nominal relative orbit configuration, the autonomous nominal regime can begin. This will involve both spacecraft enabling their autonomous FM safeguards to allow for safe proximity operations and execute formation control maneuvers on their own. In this regime, the formation is capable of autonomously transitioning between its nominal modes (Standby to Transfer to Science and back out) as it reconfigures its relative orbit. The formation will remain in this regime, maintaining the same formation roles that were set for each spacecraft in the ground controlled regime until a fault is detected or the ground commands the formation back to the ground controlled regime. If all goes well on orbit, VISORS should spend the majority of its time in this operating regime, and the autonomous formation control will help reduce mission operator workload.

Finally, the autonomous off-nominal operating regime is intended to allow fault responses to be executed according to the multi-agent fault handling strategy to and safe the formation until the ground can get involved and further diagnose the issue. In this regime, formation roles for VISORS can only be passive-passive, active-passive, or passive-active. This ensures that roles can be switched to resolve faults (even if to switch both spacecraft to passive) without ever risking both spacecraft switching themselves to active. If multiple faults are detected simultaneously, the response prescribed will address the most urgent fault that could cause the greatest risk of an intersatellite collision. Once handled, if another anomaly occurs while still in this regime, the fault detection and response logic will only prescribe a response to it if it is considered a "higher priority" fault. This is done to ensure that responses executed for low level faults do not jeopardize the effectiveness of more important faults that could pose a more significant collision risk.[17] Once this regime is entered, the ground will be alerted as soon as possible, and they can either respond by commanding the formation into the ground controlled regime (for example in the event that a CAM has been performed and the spacecraft have separated sufficiently) or allow the formation to remain in the autonomous off nominal regime until the ground has a better understanding of the situation.

VI. Implementation in the VISORS Mission's Flight Software

Effectively implementing the FM architecture described above as robust flight software presents a few challenges related to software complexity, the short development time available to the team (around 1 year), and robustness.[18] Therefore, the team focused on writing the simplest possible software solution that implementations the architecture's key elements: distributed characterization of formation health through continuous inter-satellite communications, adherence to the principles of multi-agent fault handling, consistent definitions of operating regimes, and cautious use of autonomy wherever necessary.[19] Additional emphasis was placed on reconfigurability of the software's behavior from the ground without having to re-flash the whole FSW executable. This is an important aspect that will help the mission operations team adjust the FSW more easily if and when unexpected circumstances arise on orbit and the FM logic needs to be slightly modified.

For VISORS, the F-Prime framework (developed by NASA's Jet Propulsion Laboratory) was chosen as the FSW framework for the HSA, since it comes with core FSW capabilities like message queues and threads, as well as several ready to use components. This allowed the team to focus on implementing the mission specific logic, instead of having to worry about re-developing the low-level aspects of a FSW application. In addition, F-Prime has flight heritage on the Mars Helicopter, is highly performant, and is well validated – all of which are important for this safety-critical implementation on VISORS.[20]

Since there are two VISORS spacecraft, each with an identical flight computer, the goal was to develop one common HSA deployment that can be run on both. In order to reduce the software development effort and allow this approach to scale to formations of many spacecraft in the future, these two deployments would be functionally identical and be able to discern which spacecraft they are running on via a file with the local spacecraft's identifier. Each deployment has a payload state machine (PSM) for controlling the local spacecraft's mission role and mode, and the FM architecture is split into two software components that interact directly with it. The fault detection (FD) component groups telemetry coming from both spacecraft into the macro level Boolean statuses that fully characterize the formation's health, and these are then sent to the fault response (FR) component that is tasked with selecting the most appropriate response.

Fault Detection Implementation

Finding a robust way of distilling the formation's overall health down to a few macro-level Boolean statuses is a challenge. For formation level faults, the formation statuses corresponding to the multi-agent fault handling strategy are used. However, some faults will show up first as a problem with a subsystem on the local spacecraft (before potentially causing a formation level issue), and so it is useful to have subsystem statuses to diagnose these types of faults as well. These subsystem statuses are split into hardware (HW) or

software (SW) errors for more granularity in describing the issue that has caused them. The list of these statuses currently used for fault diagnosis on VISORS are shown in Table 3 below.

Table 3. List of statuses used to represent the local's and formation's health.

Subsystem Statuses (for local)	Formation Statuses (for whole formation)
Prop HW Error	FM App Not Running (Local)
Prop SW Error	ISL Disconnected
ISL HW Error	FM App Not Running (Remote)
ISL SW Error	Nav & Guidance Not Working (Local)
Science Instrument HW Error	Nav & Guidance Not Working (Remote)
Science Instrument SW Error	Prop & Control Not Working (Local)
	Prop & Control Not Working (Remote)

The FD component needs to be able to read all the relevant telemetry channels and categorize their values to determine when and what statuses to diagnose. As shown in Table 4, each telemetry channel name (or ID number) can be linked to a subsystem or formation status with a combination, logical expression, and persistence duration. The live value being read from the telemetry channel is compared with a pre-defined threshold value via a logical comparison, and if it evaluates to true for longer than the persistence duration, the corresponding status will be diagnosed as having an error. Additionally, multiple telemetry channels can be ANDed or ORed together to contribute to diagnosing certain statuses. Using the example of Table 4 below, if the temperature of the prop system is greater than 50 C for one minute or the pressure of the prop system is greater than 5 atm for 1 minute, then a Prop HW Error will be diagnosed. It is important to note that since all subsystems except for the science payload are considered critical for maintaining safe formation flight, a subsystem error diagnosis for any of them will usually result in a formation error diagnosis if the subsystem effort is not able to be fixed in time. For example, as illustrated in Table 4, if the local spacecraft does not receive a message over the ISL from the remote for longer than two minutes or a previous ISL HW Error has been diagnosed for more than two minutes, then the ISL Disconnected formation status error will be diagnosed.

Table 4. Subsystem and formation statuses diagnosis tables with sample entries.

Telemetry Channels	**Combinations**	**Logical Comparison**	**Persistence Duration**	**Subsystem Statuses**
Temperature of prop system	OR	> 50 C	1 min	Prop HW Error
Pressure of prop system	OR	> 5 atm	1 min	
…				
Telemetry Channels	**Combinations**	**Logical Comparison**	**Persistence Duration**	**Formation Statuses**
Time since last message from remote	OR	> 2 mins	-	ISL Disconnected
ISL HW Error from Subsys FD	OR	TRUE	2 mins	
…				

Figure 10 illustrates the flow of information relating to FD and FR across the formation during nominal and off-nominal operations. A telemetry database on each spacecraft stores live telemetry values coming in from across the formation: from local subsystems, the BCT bus, other software components within the HSA itself, and statuses coming from the remote spacecraft over the ISL. The FD component then periodically reads all the relevant channels in the database at a nominal rate of 1Hz to verify if any of these values meet

the threshold conditions for error diagnosis (such as those highlighted in Table 4). After this, the relevant telemetry channels that need to be sent to the remote spacecraft to inform it of the local's formation statuses are packaged up and sent over the ISL. During nominal operations, both FD components on each spacecraft are simultaneously reading from their telemetry database, packaging up telemetry channels for the remote, and sending them to each other over the ISL, effectively closing a continuous communication loop between both spacecraft. If a fault occurs, then the FD component will diagnose it pass it on to the FR component through an FPrime port call, as well as also send the diagnosis to the remote spacecraft. After the FR component has chosen a response, it will send it to the PSM to actually execute this response. As previously discussed, the response can involve only local changes, or it could also involve a role switch command which gets sent from the local PSM to the remote PSM over the ISL.

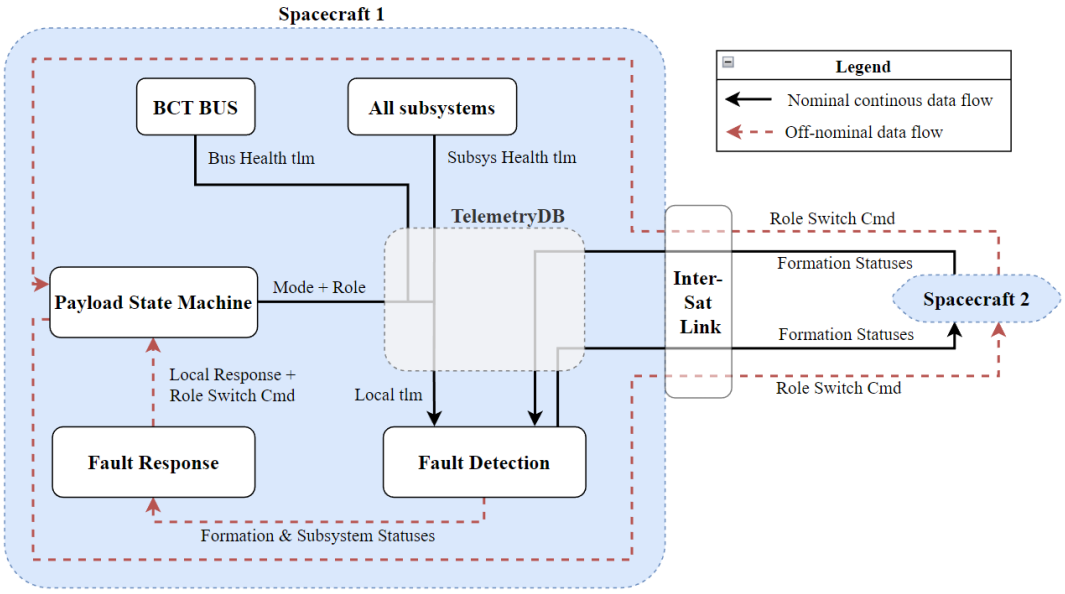

Figure 10. Information flow across the formation.

There are two additional aspects of the FD component's interaction with the FR component that must be discussed. First, the FD component continually checks for formation level "congruency" in critical health data shared between both spacecraft like the formation roles and formation status diagnoses. Each time the FD diagnoses a group of formation statuses (which happens on call of the component at a rate of 1Hz, even if the diagnosis says that all formation statuses are nominal), the local verifies that its knowledge of the remote's formation role and formation statuses matches the remote's actual role and statuses that have just been received over the ISL. Both spacecraft are continuously checking these values for congruency, and if they do not match, the FSW App will throw a software assert and restart. Such a response is warranted because the multi-agent fault handling strategy relies on the formation members having accurate information regarding each other's health as one of its main principles. If this information does not match, there may have been a bit flip or data corruption, in which case it is best for the FSW App to restart and reinitialize instead of acting on this incorrect information. Second, the FD component is implemented such that it will only pass a set of subsystems and formation statuses to the FR component if 1) one of these statuses indicates an error, and 2) the status values are not exactly the same as the status values on the previous call of the FD component. This behavior means that the FD component only calls the FR component once for each possible contingency scenario, and prevents the same diagnosis being sent multiple times in a row. If the diagnosis changes, then the FR component will be called again to recommend another response. This allows the FR component's logic to be simplified, since it doesn't have to deal with the same fault occurring multiple times (if it does, it will have been handled by the FR component the first time around).

Fault Response Implementation

After a fault diagnosis is passed onto the Fault Response component, it must choose the best response to be executed given the subsystem and formation statuses, current role, and current mission mode. This is done by selecting values for four response "recommendations" that are sent to the PSM to be executed:

1. Power cycle or turn off a local subsystem (or not).
2. Switch the local role (or not).
3. Switch the local mission mode (or not).
4. Send a role switch command to the remote as part of the active role delegation process (or not).

The main strategy in choosing which responses to recommend follows from a fairly common principle in spacecraft fault response: attempting to resolve the fault as "locally" as possible, and escalating to the next level of mission scope if this does not fix the issue. Figure 11 illustrates how the FR component applies this principle: first the fault is handled at the at the L1 subsystem level before escalation to the L2 formation level, in order to minimize the impact on the formation's operations. Handling a fault at the L1 level refers to the subsystem statuses that are diagnosed by the FD component and the power/cycle turn off response prescribed by the FR component. Depending on the issue, it may be fixed at the L1 level (i.e. a software reset) or it may not be fixed (i.e. turning off the subsystem due to an overcurrent) – in which case it will propagate up to the L2 level. At this level, a fault could be resolved by switching formation roles or modes according to the multi-agent fault handling strategy. Finally, L3 faults that involve an immediate formation safety issue (such as an imminent collision due to a passive safety violation) are always handled separately and with first priority. The L3 level of faults is kept separate from the other levels to prevent a situation where the switching or roles or modes nullifies a spacecraft's ability to perform a CAM.

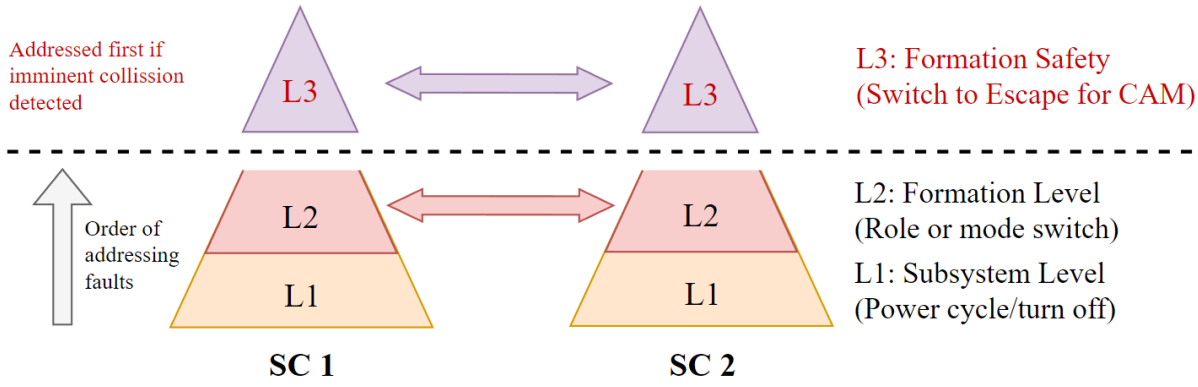

Figure 11. Key elements of fault response implementation.

The elements of FR that deal with L1, L2, and L3 responses are implemented separately in their own logic blocks for simplicity, as shown below in Figure 12. Once the FR component has received these statuses, it first executes the L1 Subsystem Response logic, which checks the statuses for each subsystem on the local spacecraft. If a HW Error is diagnosed (indicating a mechanical/electrical issue with the subsystem), it recommends a response to turn off that subsystem to prevent further damage, and if a SW Error is diagnosed (indicating a hung/unresponsive software error) it recommends a response to power cycle that subsystem in the hope of clearing the fault. There is a counter that increments each time the same subsystem is power cycled during the off-nominal autonomous regime, ensuring that a subsystem is never power cycled more than a few times during the autonomous off-nominal operating regime (counter is reset upon entry into the ground controlled regime). It is important to note that only one level of response (L1-3) should be prescribed each time the FR component is called. If the fault is not resolved with a L1 (subsystem) level response, then the FD component will re-diagnose the fault as an L2 (formation) level fault to be addressed by the FR component again, but this time with a formation level response. This is the reason for the "conditional logic" dotted line in the diagram; if an L1 level response is recommended, then the L2 block will be skipped, but if not, then the L2 block will be run to determine whether or not an L2

response is recommended. As previously mentioned, L3 responses are treated separately, so that logic block is always executed. The L3 Escape Decision Logic is simple, and involves recommending a CAM and switch to Escape mode if an imminent collision/passive safety violation is detected, and no L3 response if not. The last logic block is performs deconfliction of the L1-L3 responses to recommended upstream, in order to ensure that only one level of response is prescribed at a time.

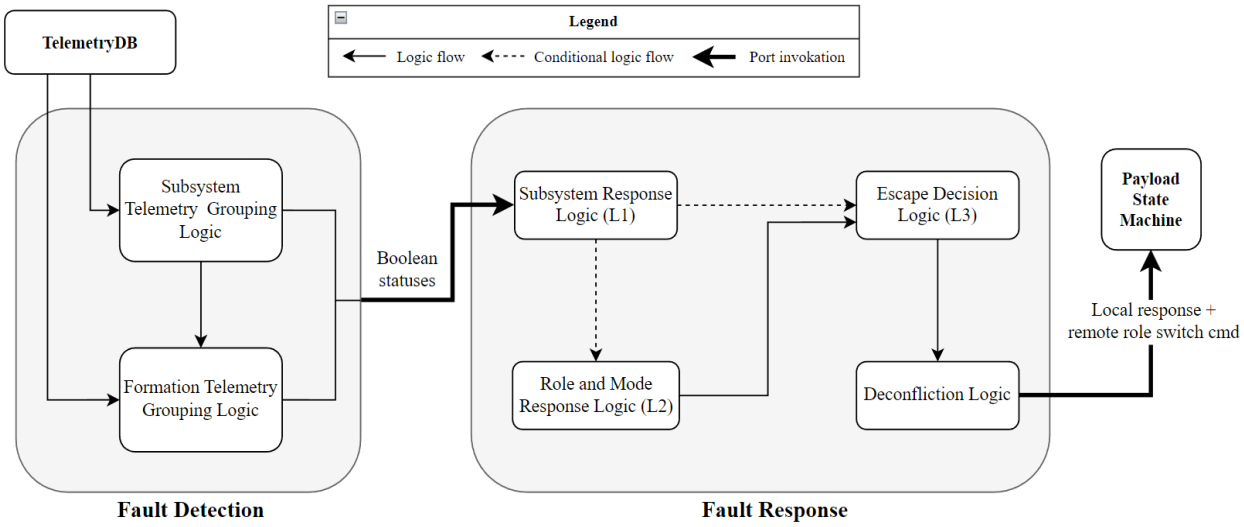

Figure 12. Close-up of internal logic blocks and interaction between fault management components.

On-orbit reconfigurability of the FD and FR logic is extremely important, due to the novelty of the mission and the uncertainty in how well many of the novel technologies enabling this concept of operations will function in space. This will allow the ground to quickly tighten or loosen the fault detection thresholds, as well as add/delete/modify specific fault responses via a simple ground command (which can be done much more readily than if an entire FSW executable re-flash was needed). In order to achieve this capability, the FD and FR logic was built around FPrime parameters – who's values can be modified via ground command. The FD channel monitors are stored in a group of parallel arrays – each one containing a vertical column in Table 4 – for each telemetry channel name/ID, combination, logical comparison, persistence value, and subsystem/formation status. These arrays are read into the FD code in parallel to actually perform the telemetry threshold monitoring and diagnosis, so uplinking new versions of these means that the ground can completely change what telemetry channels are being monitored and what their fault diagnosis thresholds are set to. Similarly, the FR component's L2 response logic from Figure 7, is actually implemented as table defined logic shown in Figure 13.

Formation Status Inputs							Response Outputs		
FM App Local	ISL with Remote	FM App Remote	Nav & Guidance Local	Nav & Guidance Remote	Control & Prop Local	Control & Prop Remote	Role Response (If Active)	Role Response (If Passive)	Mode Response
0	-	-	-	-	-	-	SendSwitch	NoSwitch	Revert to Standby
1	0	-	-	-	-	-	NoSwitch	NoSwitch	Revert to Standby
1	1	0	-	-	-	-	NoSwitch	RecvSwitch	Revert to Standby
1	1	1	0	0	-	-	NoSwitch	NoSwitch	Switch to Safe
1	1	1	1	0	-	-	NoSwitch	RecvSwitch	No mode switch
1	1	1	0	1	-	-	SendSwitch	NoSwitch	No mode switch
1	1	1	1	1	0	0	NoSwitch	NoSwitch	Switch to Safe
1	1	1	1	1	1	0	NoSwitch	RecvSwitch	No mode switch
1	1	1	1	1	0	1	SendSwitch	NoSwitch	No mode switch
1	1	1	1	1	1	1	NoSwitch	NoSwitch	No mode switch

Figure 13. Table defined logic that implements the L2 multi-agent fault handling strategy.

In the table, the inputs are the formation statuses, and the outputs are the role responses (which are different depending on the local's current role) and the mode response. The value of each formation status is compared to the table's binary values (1 means that the status is good, 0 means that the status has an error, and (-) means that the status is irrelevant). As previously discussed, not every permutation of the seven statuses needs to be considered, because once some statuses are diagnosed as having an error, the values of the other statuses become irrelevant. The row which has all 1's and 0's matching will be selected by the FR component, and the corresponding recommended role and mode switch will be chosen. Since this table is also stored as parallel arrays that are FPrime parameters, the ground can modify a responses in-flight by either changing the values of the status inputs needed to trigger a particular response (left side of the table) or changing the role and mode responses themselves (right side of the table).

VII. Future Work: Validation and Testing Methodology

A validation campaign for this VISORS FM software implementation is currently underway, involving extensive unit testing and integration testing. As of the writing of this paper, the PSM, FD, and FR components have been written and fully unit tested, but the majority of the flight software's integration testing remains, including:

- Verifying that all software components in the HSA deployment interface with each other correctly and that it can be configured via commands as expected.
- Verifying that two HSA deployments connected together on a local computer over a simulated ISL are able to continuously exchanging data back and forth and effectively demonstrate the multi-agent fault handling strategy.
- Profiling the finished code to ensure that it meets program/data memory requirements and there are no memory leaks.
- Verifying that the HSA runs properly on the flight computer, interacts as expected with the BCT FSW, and tuning the FD thresholds before flight based on real subsystem telemetry.

The remaining portions of this testing campaign have already been thoroughly planned out and will be carried out in time for a final VISORS FSW delivery date of Q4 2023.[21] Additional details regarding testing results from items in the list above will be included in a paper submitted to the Small Satellite Conference 2023.

VIII. Conclusion

In conclusion, this paper presents a detailed look at the failure analysis, design of the FD/FR elements, and software implementation of a novel FM architecture for spacecraft formations involving proximity operations. This architecture is based on the premise of a well-designed relative orbit that includes a sufficient passive safety margin as well as the ability to perform a CAM. An FMECA focused on the inter-satellite collision system failure was performed to highlight the temporary nature of FFFs faults that can have serious consequences during proximity operations. These risks were effectively mitigated by devising an elegant multi-agent fault handling strategy that relies on timely fault detection and information sharing across the formation, and a coordinated approach to fault response enabled by the redundancy of a homogenous formation. In order to safely utilize autonomy for this FM architecture, it was found that strict adherence to the flight rules under each operational regime was necessary. Finally, a sample software implementation of this architecture was demonstrated, making heavy use of reconfigurable parameters to allow operational flexibility in modifying the FD and FR software's behavior on orbit.

The pioneering VISORS formation flying mission is used as a case study throughout this paper in how this FM architecture can be tailored to a specific mission involving two small satellites equipped with a certain set of navigation and ISL technologies. However, this architecture is not confined to formation with a specific number of spacecraft or the same formation-enabling technologies VISORS has. Instead the same core principles described in this paper lend themselves to other kinds of distributed space systems where an inter-satellite collision is a primary mission risk.

IX. Acknowledgements

I would like to express my gratitude towards everyone who has supported and encouraged me over the last six years at Georgia Tech, making them the most challenging, enlightening, and fulfilling years of my life. To my mom, Sherilyn Krizmanic, for making me believe in my own capabilities and constantly pushing me to be the best version of myself; I would be nowhere without you. To my dad, Yves Paletta, for teaching me the value of consistency and discipline, and being a role model in how to live a balanced and purposeful life. To my sister, Marie Paletta, for loving me unconditionally and showing me the value of living life to the fullest. To my extended family for always celebrating my achievements and providing encouragement. To all my friends and mentors at my various internships and school endeavors like RASC-AL, Design Build Fly, and the Astronomy club. To my roommates Tristan, Paul, Gabe, Rishab, Nilabh, Rony, and Harrison for being there for me and being awesome to live with. To the Georgia Tech VISORS team Ebenezer, Will, Grace, Kevin, Lizzie, Joseph, Donald, Spencer, Justin, Henry, Shan, Sam, Bryan, Michael, Mark, Kian, Adit, and Alec for all their contributions to this incredible project, for the creation of a hardworking yet relaxed team atmosphere, and for shared highs and lows. And to all my peers in the Space Systems Design Laboratory who have created such a welcoming and supportive academic and professional community.

I would also like to thank the entire VISORS team: Farzad, Phil, Jake, Adrian, Doug, Tommaso, Toby, Mason, Samuel, Adam, Annika, Om, Petr, and many others too numerous to list here for their continued efforts in bringing this groundbreaking mission from concept to reality. It has been a pleasure and a privilege to work amongst such a talented and supportive team.

Lastly, to Dr. Glenn Lightsey: thank you for giving me the opportunity to work on incredible projects within an amazing community of students at the SSDL, for providing resources and freedom which have allowed me to grow as an engineer and a leader, for helping me acquire the export control license that has allowed me the wonderful opportunity to deeply contribute to VISORS, and for offering advice, encouragement, and support through my years at Georgia Tech.

Some of the material in this paper is based upon work funded by the National Science Foundation under grant No. 1936576. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

X. References

[1] Melissa R. Jones, K. A. (2019). *Integrating Reliability Engineering with Fault Management to Create Resilient Space Systems.* Laurel, MD: John Hopkins Applied Physics Lab.

[2] Tomita, K., Shimane, Y., & Ho, K. (2022). Small Body Reconnaissance by Multiple Spacecraft Via Deep Reinforcement Learning. *Proceedings of AAS/AIAA Astrodynamics Specialist Conference.* American Astronautical Society.

[3] Galchenko, P., & Paletta, A. (2022). Attitude Determination and Control System Design with Orbit Considerations for the GTOSAT Mission. *Small Satellite Conference.* Logan, UT: Utah State University.

[4] Galchenko, P., & Paletta, A. (2023). Attitude Determination and Control System Design for the GTOSat Mission. *AAS Guidance Navigation and Control Conference.* Breckenridge, CO: American Astronautical Society.

[5] Ku, E., & Paletta, A. (2020). Mission for an Impermanent Surface Stay to Investigate Our Neighbor, Mars. *AIAA ASCEND.* AIAA.

[6] John Day, M. I. (2011). *Fault Management at JPL: Past, Present and Future.* Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

[7] McDonald, D. (2022). Fault Management in Small Satellites. *Master's Report.* Atlanta, GA: Georgia Institute of Technology.

[8] Koenig, A. W., D'Amico, S., & Lightsey, E. G. (2021). Formation Flying Orbit and Control Concept for the VISORS Mission. *SciTech 2021 Forum* (pp. 2021-0423). San Diego: AIAA.

[9] Koenig, A., & D'Amico, S. (2020). *GNC Safety Plan for the VISORS Mission.* Palo Alto, CA: Stanford University.

[10] Sasaki, T., Ho, K., & Lightsey, E. G. (2022). Nonlinear Spacecraft Formation Flying using Constrained Differential Dynamic Programming,. *Proceedings of AAS/AIAA Astrodynamics Specialist Conference.* American Astronautical Society.

[11] Hart, S. T., Daniel, N., Hartigan, M., & Lightsey, E. G. (2022). Design of the 3-D Printed Cold Gas Propulsion Systems for the VISORS Mission. *Proceedings of AAS/AIAA Astrodynamics Specialist Conference.* American Astronautical Society.

[12] Lightsey, E. G. (2022). CONCEPT OF OPERATIONS FOR THE VISORS MISSION: A TWO SATELLITE CUBESAT FORMATION FLYING TELESCOPE. *AAS Guidance, Navigation, and Control Conference.* Breckenridge, CO: American Astronautical Society.

[13] Paletta, A. (2022). Development of a Contingency Operations Architecture for the VISORS Formation Flying Space Telescope. *Small Satellite Conference.* Logan, UT: Utah State University.

[14] Lindsey, N. J. (2016). *An Innovative Goddard Space Flight Center (GSFC) Methodology for using FMECA as a Risk Assessment and Communication Tool .* Greenbelt, MD: NASA Goddard.

[15] Dennehy, C. J., & Carpenter, J. R. (2011). *A Summary of the Rendezvous, Proximity Operations, Docking, and Undocking (RPODU) Lessons Learned from the Defense Advanced Research Project Agency (DARPA) Orbital Express (OE) Demonstration System Mission.* Hampton, Virginia: National Aeronautics and Space Administration.

[16] Hauge, M. (2023). Operations Systems Engineering for the Lunar Flashlight Mission. *Master's Report.* Atlanta, GA: Georgia Institute of Technology.

[17] Tomita, K., Skinner, K. A., & Ho, K. (2022). Bayesian deep learning for segmentation for autonomous safe planetary landing. *Journal of Spacecraft and Rockets*, 1800-1808.

[18] Arunkunar, E. (2023). Design of the Hosted Software Application for the VISORS Mission. *Master's Report.* Atlanta, GA: Georgia Institute of Technology.

[19] Paletta, A., & Arunkumar, E. (2023). Development of Autonomous FDIR Logic to Enable Safe Prox Ops for the VISORS Mission. *JPL Flight Software Workshop.* Los Angeles, CA: NASA JPL.

[20] Anderson, J. (2023). Upgrading Ingenuity's Flight Software. *JPL Flight Software Workshop.* Los Angeles, CA: NASA JPL.

[21] Paletta, A. (2023). Testing Methodology For Spacecraft Precision Formation Flying Missions. *AAS Guidance, Navigation, & Control Conference.* Breckenridge, CO: American Astronautical Society.