SSC23-XI-06

Development of an Autonomous Distributed Fault Management Architecture for Spacecraft Formations Involving Proximity Operations

Antoine Paletta, Ebenezer Arunkumar, Glenn Lightsey
Georgia Institute of Technology
625 Cherry St NW, Atlanta, GA, 30318; +1 6784698197
apaletta3@gatech.edu

ABSTRACT

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 strategy. The primary focus is to enable the reliable detection of faults occurring on any formation member in real time and the autonomous decision making needed to resolve them while keeping 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 solar telescope, 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 strategies, and the flight software implementation for VISORS are discussed in the paper.

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 a mission's 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/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 traditional 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 a backup system or putting the spacecraft into a safe 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 depends on the spacecraft's fault tolerance capabilities and the mission's operational constraints.[1]

Background on Spacecraft Formation Flying and Proximity Operations

As space missions become increasingly complex, the need for more affordable ways to execute them grows. One potential way of reducing cost for more complex missions is through a distributed spacecraft system (DSS), which consists of multiple spacecraft working together to achieve mission objectives that would otherwise be significantly more expensive or infeasible to perform with a single spacecraft. 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 inter-satellite 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 approach to fault management, often described as more of an art form than a science.[2] Therefore, 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. 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.

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 PFF missions including intersatellite links, autonomous relative maneuver planning and control, and miniaturized on-board propulsion.

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 as it takes multiple images.

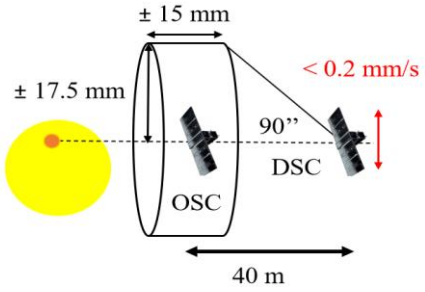

Figure 2. Relative position/velocity requirements imposed on formation during observation.

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: if an imminent inter-satellite collision is detected, a collision avoidance maneuver (CAM) designed to increase the inter-satellite separation can be performed.

This paper will discuss how the passive and active orbital safety elements are leveraged to significantly reduce the risk of a collision on orbit. Since the along-track separation between both spacecraft is subject to high uncertainty under the effects of differential drag and orbital perturbations, a sufficient passive safety margin is achieved using a relative orbit design called “eccentricity/inclination vector separation”.[3] This method, resulting in a relative orbit shown in blue 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

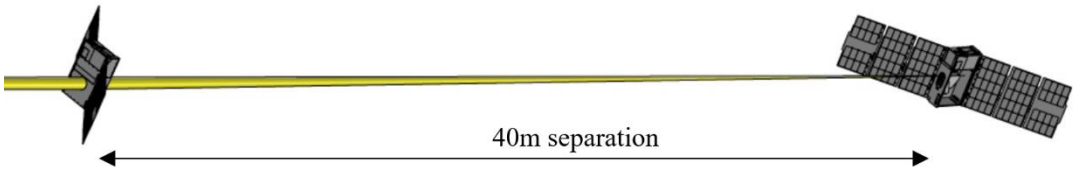

Figure 1. VISORS formation during observation.

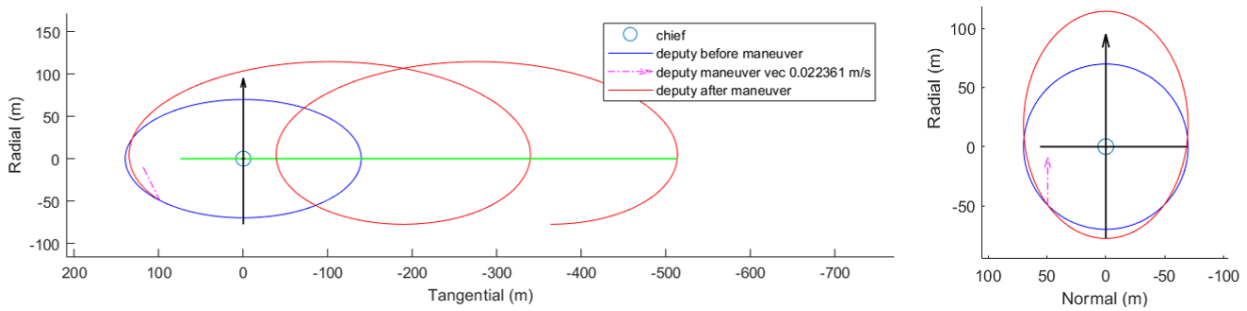

Figure 3. Illustration of relative motion before (solid line) and after (dashed line) a collision avoidance maneuver in the RTN frame.

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.[4]

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.[5] 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. Therefore, from a formation control perspective, both spacecraft are identical, 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 sun-synchronous LEO 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, and a hosted software application (HSA) that lives on a partition of the BCT Xilinx flight computer.[6]

The HSA contains 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 state machine/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 exists in the "hosted software payload" paradigm where it is only alive and running when the BCT bus is in its Fine Reference Point (FRP) mode, as shown in Figure 4. When the BCT bus is in its Launch, Sun Point, and Survival modes, the HSA is turned off. Additionally, Figure 4 shows how there are no autonomous transitions into FRP, only transitions out of it. This means that when the BCT bus's own autonomous fault management system detects a bus-level fault (such as an undervoltage, invalid attitude, etc.) and transitions to Sun Point mode, it will stay there with the HSA off until the ground can establish contact and command a transition back into FRP mode.

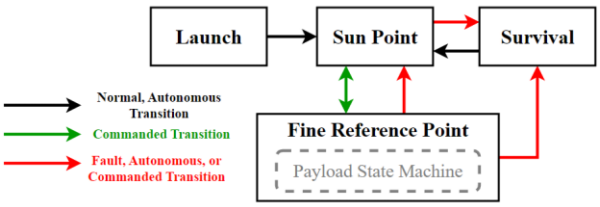

Figure 4. BCT bus mission modes.[7]

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).[8] A proximity operations mission such as VISORS differs compared to a typical single spacecraft mission as there are two different types of total system failures possible. The first kind – referred to here as a premature end of mission (P-EOM) – exists for every space mission and would likely result from permanents faults in the subsystems on either spacecraft. The second kind, – referred to here as an inter-satellite collision end of mission (ISC-EOM) – can be caused by certain combinations of temporary faults in the subsystems involved in formation control on either spacecraft. In terms of risk mitigation, the P-EOMs are addressed by having redundancy in certain subsystems (i.e. both spacecraft have a functionally identical propulsion system). However, the ISC-EOM is best addressed with operational mitigation strategies built into the software and orbit design. The fault analysis and the rest of the paper will focus on addressing the ISC-EOM.

In order to safely control the formation with both spacecraft in such close proximity, certain functionalities (referred to here as Formation Flying Functionalities or FFFs) shown in Table 1 below must be continuously operating nominally. 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 ISC-EOM. The severity of the collision risk posed is related to the specific combination of faults occurring across the formation and their duration.

Table 1. Formation Flying Functionality Descriptions.

FFF	Description
FSW Application	The FSW application (the HSA for VISORS) needs to be running in order to manage the formation flight, the payload's states, and any faults that may occur.
Inter-sat Link	The ISL needs to be working in order for multiple spacecraft to coordinate with each other and exchange relevant data.
Maneuver Planning	The relative navigation filter needs to be converged in order to generate precise state estimates that can be used to predict the formation's motion, and the guidance algorithms must be able to autonomously plan maneuvers to maintain the relative orbit.
Maneuver Execution	At least one spacecraft needs to be capable of performing the planned maneuvers to control the relative orbit.

Risk Categorization

In order to categorize the different risks and perform an effective fault analysis, all the possible combinations of FFF faults are assigned to an ISC-EOM scenario and 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.[9] 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 ISC-EOM is shown below in Figure 5. When performing this analysis, it is important to evaluate the metrics described above from the perspective of a spacecraft's FSW App, acknowledging the limitations in perfect awareness of the rest of the formation. In some scenarios, faults may be occurring on the spacecraft in question (referred to as the "local"), in others they may be occurring on the adjacent spacecraft (referred to as the "remote"), and in others yet they may be occurring on both spacecraft simultaneously. The presence of a fault in one FFF often cascades down into other FFFs, which means that only about ~15 scenarios need to be considered in this FMECA matrix. For example, if the FSW App is not running, none of the other FFFs can function, or if the maneuver planning is not working, then maneuver execution capability doesn't matter.

As can be seen from the FMECA matrix, the highest RPNs indicate the most important scenarios to mitigate. Highlighting some examples, scenarios #4 and #5 are concerning as they would result in a high collision risk, are likely to occur at least once during the mission lifetime (due to the ISL subsystem not having flight heritage), and would be difficult to diagnose due to the lack of communication between both spacecraft. Scenario #13, which represents a sudden transition out of BCT's FRP mode and subsequent turning off of the VISORS payload on the active spacecraft, is similarly concerning. In this case, the severity is high as the active spacecraft is no longer be able to maneuver, the likelihood is high due to the prevalence of Single Event

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

Upsets (SEU), and the observability is low as the passive spacecraft would suddenly stop hearing from the active with no warning, resulting in an uncontrolled formation.

MULTI-AGENT FAULT HANDLING STRATEGY

A multi-agent fault handling strategy was devised to handle the concerning ISC-EOM scenarios highlighted above. This strategy leverages the inherent redundancy that exists in over-actuated (from a robotics perspective) spacecraft formations like VISORS. 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 the underperforming/failing spacecraft(s). There are two desirable 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 PFF mission that detects faults and safes the formation before the ground has a chance to react. 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. This multi-agent fault handling strategy can be applied to PFF missions with arbitrary numbers of spacecraft, and as such its key elements will be discussed more generally.

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 with all other members. In this context, the SOH of each spacecraft will be defined as macro-level statuses that correspond to the FFFs. These are as follows:

1. FSW Application Running?
2. ISL Connected to other members?
3. Maneuver Planning Working?
4. Maneuver Execution Working?

If all four of these *formation statuses* are true on the local spacecraft, then it is considered perfectly healthy and can contribute to active formation control. If some or all the formation statuses on the local are false, then its ability to contribute to active maintenance of the formation could be hampered, depending on which statuses are false.

An example of this distributed formation health characterization applied to an arbitrary 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

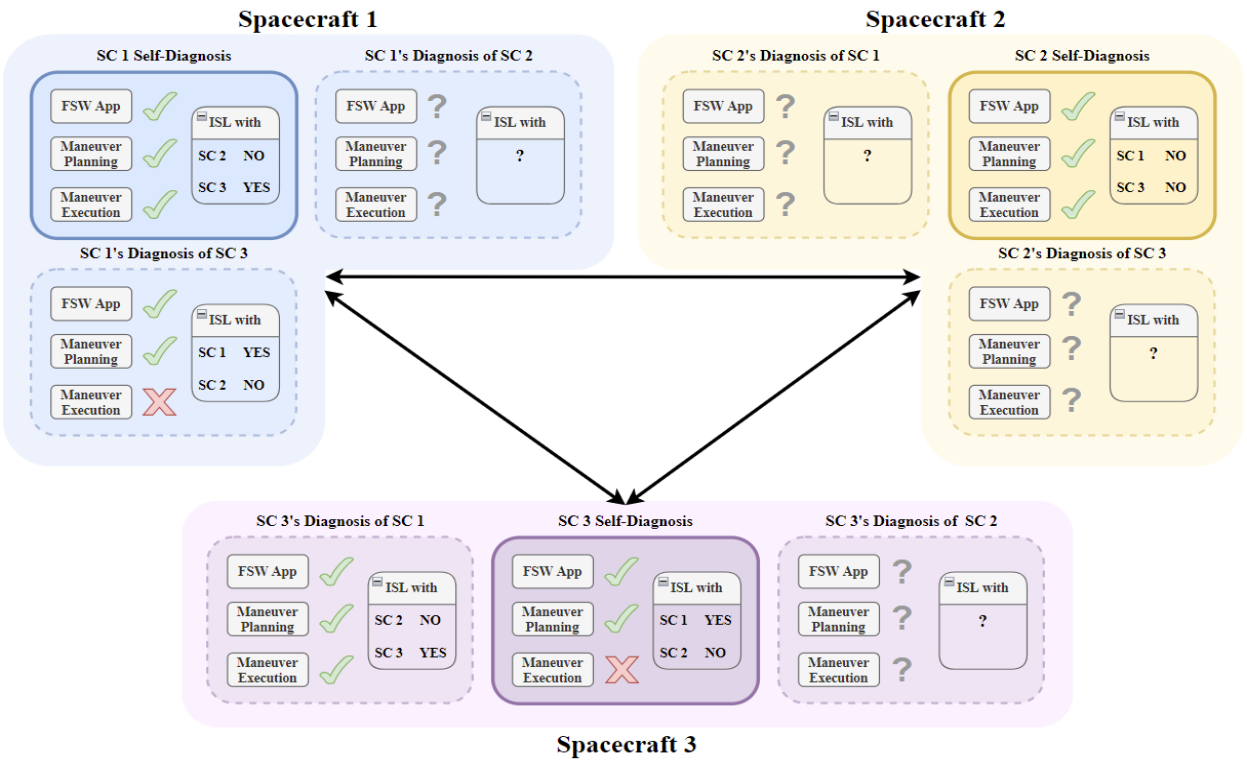

Figure 6. Formation statuses sharing scheme across a formation of three spacecraft.[10]

connected with any other spacecraft, and spacecraft 3's maneuver execution is not working. 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. However, due to spacecraft 2's malfunctioning ISL it is not able to connect to spacecraft 1 or 3, and so can only store an "invalid" diagnosis for spacecraft 1 and 3. Similarly, spacecraft 1 and 3 store an "invalid" diagnosis of spacecraft 2. Temporary outages in inter-satellite communications are 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 considered 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 shown in Table 2. If any of these assumptions are violated, the formation status sharing scheme can no longer be considered valid and it is not advisable for a response to be executed autonomously. It is important that this fault response strategy be implemented in such a way that a deterministic outcome can be predicted for any possible contingency scenario, as there are multiple examples of rigid autonomous logic behaving in unpredictable ways on past proximity operations missions because logic was autonomously executed even though its underpinning assumptions had been violated.[11]

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

Principle	Rationale
1. Each member will attempt to determine if it is healthy.	Each member of the formation must continuously be evaluating its health so that if an anomaly occurs with one member, it is straightforward to decide which other member is best suited to help take over for the failing member.
2. If a member is not receiving remote SOH information, it cannot execute any formation level responses.	It is better to not act than to act on incorrect or out of date information about the rest of the formation.
3. If a member is receiving remote SOH information and is healthy, it will assume responsibility execute any formation level responses.	Given principle #2, connected members must assume the responsibility for responding to faults.
4. If two members are temporarily not connected with an ISL, both must assume responsibility for the blackout.	There is no reliable way to tell if a link issue is on the local or 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 responsibility for a blackout.

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 concurrently when they are not supposed to. On VISORS, it ensures that 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 demonstrates how to apply the multi-agent fault handling strategy across a formation of two spacecraft like VISORS to choose a formation level response. This algorithm is intended to be run on each spacecraft in the formation, whenever a fault has been diagnosed. There is a linear progression through the 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 transfer out to a larger holding relative orbit to slowly begin to increase spacecraft separation, a continuation of nominal operations, and a cessation of formation control (local spacecraft stop maneuvering). 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, a currently passive spacecraft receiving that role switch command to promote itself, 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 necessitates a CAM to be performed. The CAM "outcome" is considered before all the others as it is the most time sensitive. The next three statuses, the FSW App Local, ISL, and FSW App Remote represent a situation where both spacecraft may have invalid SOH information about each other, warranting a conservative response. If the remote spacecraft stops communicating during the mission with no warning, it could either be due to an ISL failure on the local/remote (principle #4), or the remote's FSW App shutting down. These two scenarios can be distinguished from one another by building functionality into the FSW App's assert handler that sends an emergency message alerting the remote that it is about to shutdown whenever a FSW assert occurs. First, if the local spacecraft stops receiving messages from the remote with no warning, the cause is almost certainly an ISL failure on either spacecraft. In this situation, the roles should not be switched (due to principle #4) and the currently active spacecraft should cautiously start increasing its inter-satellite separation based on its last known state of the remote. Second, if the local stops receiving messages from the remote, and a shutdown message is the last message to be received, the cause is almost certainly due to the remote's FSW App shutting down. In this case the spacecraft at fault is the remote, and the role should be switched as necessary to ensure that the properly functioning spacecraft is currently active.

If the FSW App Local/Remote and ISL statuses are all nominal, then the next group of four statuses (Maneuver Planning Local/Remote and Maneuver Execution Local/Remote) can be considered. Due to principles #1, #2, and #3, it can be assumed that both spacecraft 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 the role should

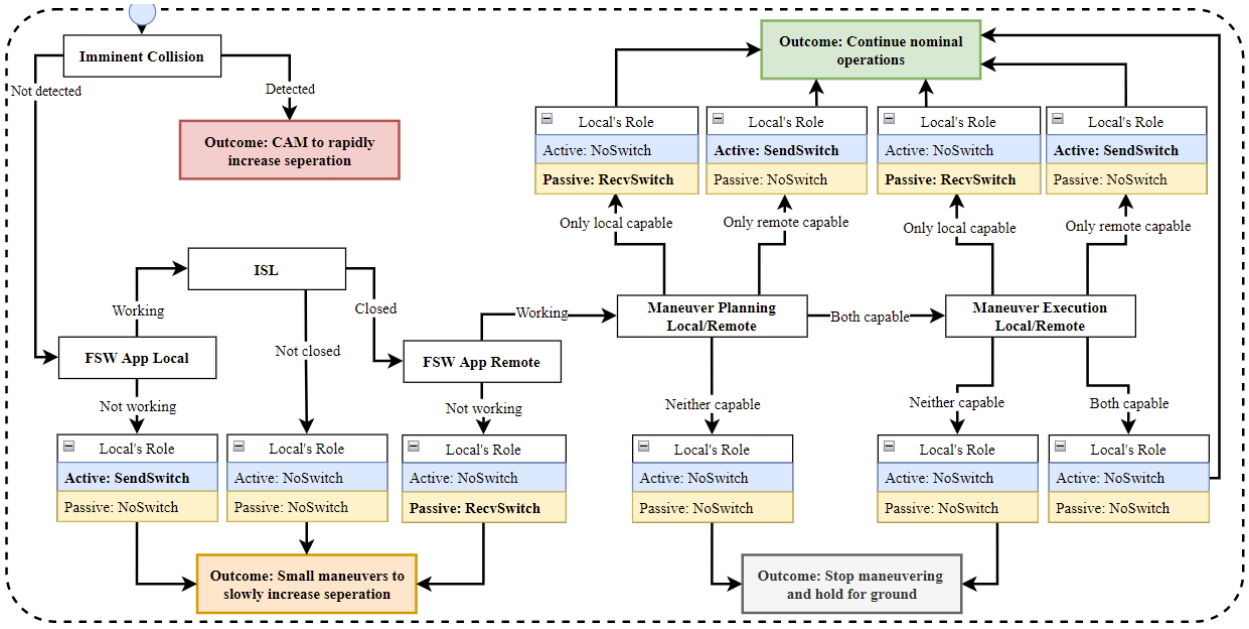

Figure 7. Decision algorithm to run on each spacecraft and illustrate how to resolve faults.

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 switching roles will not solve the issue. The "outcome" here is that both spacecraft stop maneuvering and wait for the ground to intervene. If both spacecraft have the same FFF capability, then again no role switching should occur.

In practice, this algorithm would be called upon to choose a response on both spacecraft nearly simultaneously but with 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 handle faults occurring across more than two spacecraft.

OPERATIONAL MODES AND REGIMES

The VISORS mission makes use of three nominal modes and two off-nominal modes to execute its concept of operations, shown in Figure 8. In addition, 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 multi-agent fault handling algorithm discussed above.

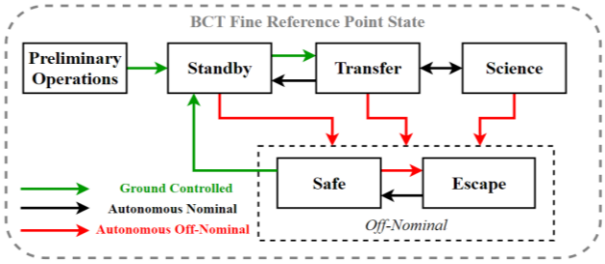

Figure 8. Payload defined nominal and off-nominal mission modes.

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 formation flying 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.[7]

The off-nominal modes for VISORS have been designed as a combination of a Safe and an Escape mode. As is normal for safe modes, this Safe mode can be entered autonomously in the event of an anomaly and acts as a holding mode where no maneuvers are allowed. It can be exited via ground command, or if a collision risk is detected and a CAM maneuver needs to be executed – in which case a transition to Escape mode occurs. Once the CAM has been executed, the spacecraft will transition back to Safe mode, with future autonomous transitions to Escape mode prevented since one CAM should be sufficient to mitigate a collision risk. While the two should always be in the same mode during nominal operations, it is possible for the modes to be different during off-nominal operations, which is accounted for.

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 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 the FSW or a mission operations team – that govern how the formation should autonomously behave and be interacted with from the ground.[16] There are three different operating regimes defined for VISORS (shown in Figure 9): a ground-controlled regime where both spacecraft are operated manually, an autonomous nominal regime where the formation is controlling itself while executing the science 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.

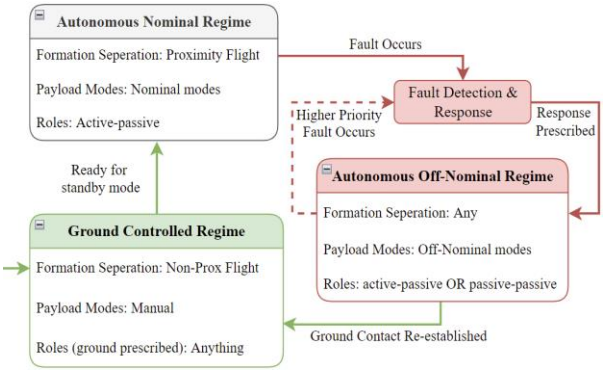

Figure 9. Three different operating regimes.

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 can 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 such 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. In this regime, autonomous maneuvers are allowed and autonomous FM safeguards are enabled. 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 with its current roles until either a fault is detected, or the ground commands it 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 autonomously executed according to the multi-agent fault handling strategy 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 previous collision avoiding responses. Once this regime is entered, the ground will be alerted as soon as possible. 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 a the root cause of the anomaly is identified.

FAULT MANAGEMENT SOFTWARE IMPLEMENTATION

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.[13] Therefore, the VISORS team focused on writing the simplest possible software solution that implements 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. 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 easily if and when unexpected circumstances arise on orbit and the FM logic needs to be modified. This software implementation also needs to handle certain common issues that don't affect the formation's health, but could damage the hardware on either of the spacecraft (similar to single-spacecraft fault management).

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.[14]

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 with arbitrarily many spacecraft, these two deployments are functionally identical; and simply able to discern which spacecraft they are running on via a local identifier. Each deployment has a payload state machine (PSM) for controlling the local spacecraft's mission role and mode,

Table 3. 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	
…				

and the FM architecture is split into two software components that directly interact 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 SOH 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. Therefore, it is useful to have a set of 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 4 below.

Table 4. List of statuses used to represent the local's and formation's health.

Subsystem Statuses (local)	Formation Statuses (whole formation)
Prop HW Error	FSW App Not Running (Local)
Prop SW Error	ISL Disconnected
ISL HW Error	FSW App Not Running (Remote)
ISL SW Error	Maneuver Planning Not Working (Local)
Science Instr. HW Error	Maneuver Planning Not Working (Remote)
Science Instr. SW Error	Maneuver Execution Not Working (Local)
	Maneuver Execution 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 3, each telemetry channel name (or ID number) can be linked to a subsystem/formation status with a combination, logical expression, and persistence duration. The live value being read from the telemetry channel is logically compared with a pre-defined threshold value, 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. This can be illustrated with an example in Table 3: if the temperature of the propulsion system is greater than 50 C for one minute, or the pressure of the propulsion 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. This can again be illustrated in Table 3: 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 will be diagnosed as disconnected.

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. 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 data for the remote, and sending them to each other over the ISL, effectively closing a continuous communication loop between both spacecraft. When a fault occurs, the FD component will create a diagnosis of subsystem/formation statuses, which is then passed to the remote spacecraft and the local's FR component. After the FR component has chosen a response, it will send it to the PSM for it to be executed. 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.

The FD component is implemented such that it will only pass a set of anomalous subsystem/formation statuses if they do not exactly match the ones from the component's previous call. This behavior means that the FD component only calls the FR component once for each unique diagnosis, and the FR component is expected to resolve the fault as best as it can. If the diagnosis changes, then the FR component will be called again to select 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.

Fault Response Implementation

After a fault diagnosis is passed onto the FR 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":

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 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 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 of roles or modes nullifies a spacecraft's ability to perform a CAM.

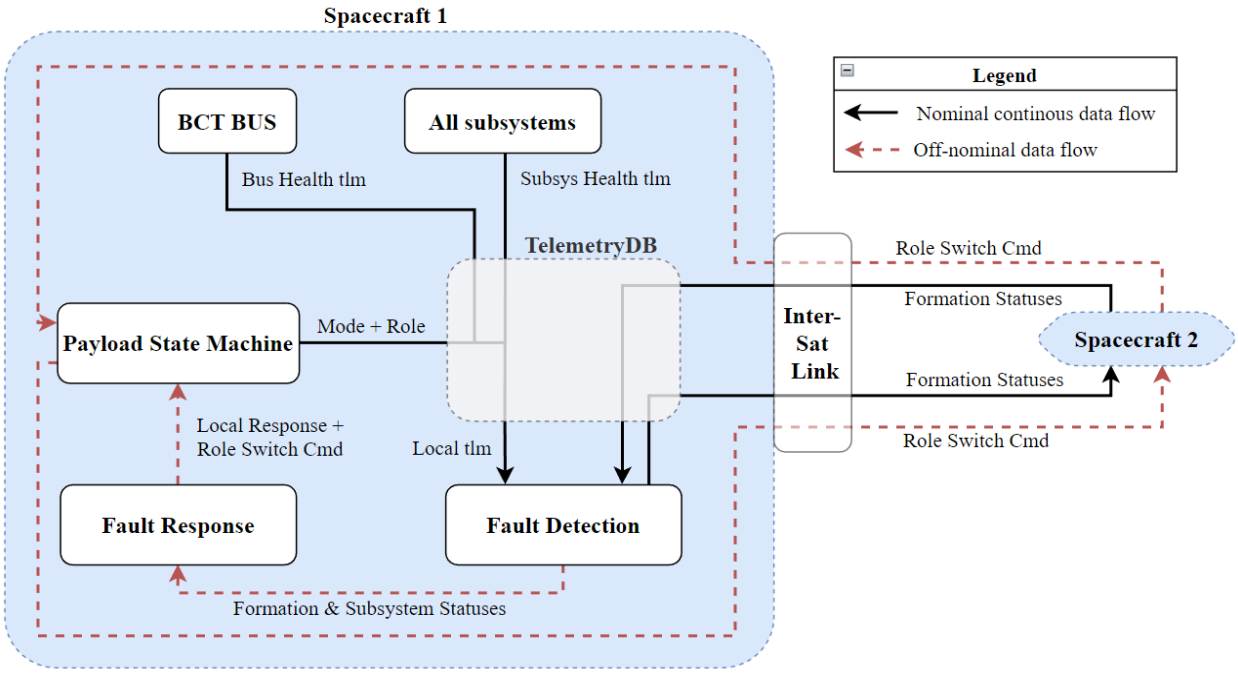

Figure 10. Information flow across the formation.

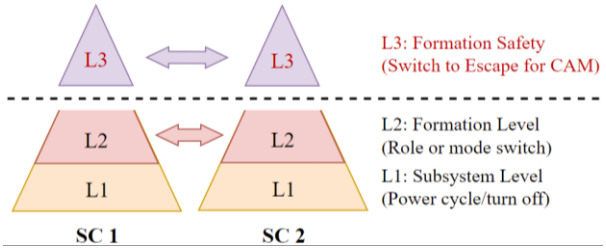

Figure 11. Key elements of FR implementation.

The elements of FR that deal with L1, L2, and L3 responses are implemented separately for simplicity, as shown in Figure 12. Once the FR component has received the Boolean statuses, it first executes the L1 Subsystem Response logic for the local spacecraft. If a HW Error is diagnosed (indicating a mechanical or 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 (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 an L1 (subsystem) level response, then on the next FD component call it 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, the L2 block will be run. The L3 Escape Decision Logic block 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 performs deconfliction of the L1-L3 responses recommended upstream in order to ensure that only one level of response is prescribed at a time.

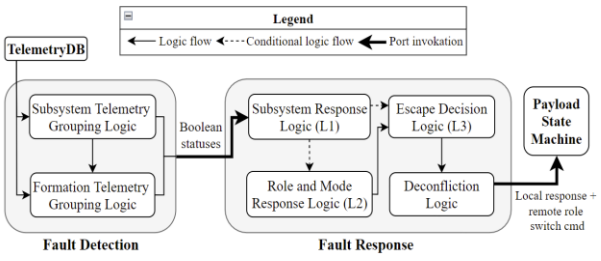

Figure 12. Close-up of internal logic blocks and interaction between fault management components.

On-orbit reconfigurability of the FD and FR logic is extremely important, due to the novelty of the VISORS mission and lack of flight heritage of some formation flying technologies on-board. 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). To achieve this, the FD and FR logic was built around FPrime parameters – non-volatile variables whose 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 from Table 3 – 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 parameters 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 implemented as table defined logic shown in Figure 13.

Formation Status Inputs							Response Outputs		
FSW App Local	ISL with Remote	FSW App Remote	Maneuver Planning Local	Maneuver Planning Remote	Maneuver Execution Local	Maneuver Execution 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 response logic.

In the table, the inputs are formation statuses, and the outputs are a role response (which are different depending on the local's current role) and a mode response. The value of each formation status is compared to the table's binary values (1 means the status is true, 0 means that the status is false, and (-) means 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 false, the values of other statuses can become irrelevant. The row which has all matching binary values 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 array FPrime parameters, the

ground can modify the L2 response logic 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).

FUTURE WORK: TESTING METHODOLOGY

A validation campaign for this VISORS FM software implementation is currently underway, involving extensive unit 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 exchange 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.[15]

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. FMECA focused on the ISC-EOM was performed to highlight the temporary nature of FFF 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, 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 once 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 formations with a specific number of spacecraft or the same formation-enabling technologies as VISORS. 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.

ACKNOWLEDGEMENTS

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.

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. John Day, M. I. (2011). *Fault Management at JPL: Past, Present and Future.* Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.
3. 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.
4. Koenig, A., & D'Amico, S. (2020). *GNC Safety Plan for the VISORS Mission.* Palo Alto, CA: Stanford University.
5. 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.
6. 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.

7. 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.

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

9. 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.

10. Paletta, A. (2023). Development of an Autonomous Distributed Fault Management Architecture for the VISORS Mission. *Master's Report.* Atlanta, GA: Georgia Institute of Technology.

11. 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.

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

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

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

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