NASA/TM-2018-219965

Spacecraft Dormancy Autonomy Analysis for a Crewed Martian Mission

Julia Badger
Lead Author, Editor

Contributors:

Avionics	*Don Higbee*
Communications	*Tim Kennedy, Sharada Vitalpur*
ECLSS	*Miriam Sargusingh, Sarah Shull*
Guidance, Navigation and Control	*Bill Othon*
Power	*Francis J. Davies*
Propulsion	*Eric Hurlbert*
Robotics	*Julia Badger*
Software	*Neil Townsend*
Spacecraft Emergency Responses	*Jeff Mauldin, Emily Nelson*
Structures	*Kornel Nagy*
Thermal	*Katy Hurlbert*
Vehicle Systems Management	*Jeremy Frank*
Crew Perspective	*Stan Love*

National Aeronautics and
Space Administration

July 2018

NASA/TM-2018-219965

Spacecraft Dormancy Autonomy Analysis for a Crewed Martian Mission

Julia Badger
Lead Author, Editor

Contributors:

Avionics	*Don Higbee*
Communications	*Tim Kennedy, Sharada Vitalpur*
ECLSS	*Miriam Sargusingh, Sarah Shull*
Guidance, Navigation and Control	*Bill Othon*
Power	*Francis J. Davies*
Propulsion	*Eric Hurlbert*
Robotics	*Julia Badger*
Software	*Neil Townsend*
Spacecraft Emergency Responses	*Jeff Mauldin, Emily Nelson*
Structures	*Kornel Nagy*
Thermal	*Katy Hurlbert*
Vehicle Systems Management	*Jeremy Frank*
Crew Perspective	*Stan Love*

National Aeronautics and
Space Administration

July 2018

Available from:

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

This report is also available in electronic form at http://www.sti.nasa.gov/ and http://ntrs.nasa.gov/

Executive Summary

Current concepts of operations for human exploration of Mars center on the staged deployment of spacecraft, logistics, and crew. Though most studies focus on the needs for human occupation of the spacecraft and habitats, these resources will spend most of their lifetime unoccupied. As such, it is important to identify the operational state of the unoccupied spacecraft or habitat, as well as to design the systems to enable the appropriate level of autonomy. Key goals for this study include providing a realistic assessment of what "dormancy" entails for human spacecraft, exploring gaps in state-of-the-art for autonomy in human spacecraft design, providing recommendations for investments in autonomous systems technology development, and developing architectural requirements for spacecraft that must be autonomous during dormant operations.

The mission that was chosen is based on a crewed mission to Mars. In particular, this study focuses on the time that the spacecraft that carried humans to Mars spends dormant in Martian orbit while the crew carries out a surface mission. Communications constraints are assumed to be severe, with limited bandwidth and limited ability to send commands and receive telemetry. The assumptions made as part of this mission have close parallels with mission scenarios envisioned for dormant cis-lunar habitats that are stepping-stones to Mars missions. As such, the data in this report is expected to be broadly applicable to all dormant deep space human spacecraft.

A functional breakdown of operations during dormancy was conducted for twelve spacecraft subsystems. Several "phases" of dormancy were analyzed, include transition operations (moving to or from crewed operations), nominal operations, contingency (emergency) operations, and preventative maintenance. Though "dormancy" is how the uncrewed period is commonly described, most spacecraft subsystems will continue to be active. During transition operations, nearly all spacecraft subsystems will be close to or at full operations during the uncrewed period for some small amount of time. Even during nominal dormancy phase, most subsystems will continue to operate in the same mode. The Environmental Control and Life Support System is an exception, due to the absence of a key component of the system, the humans. Also, the criticality of faults is reduced for many systems during uncrewed phases, which may change failure response times and functions.

In addition to the operations analysis, a study of the drivers for autonomy was conducted based on the mission parameters and an assessment of the state of the art. While several autonomy drivers were found, the two that had the most influence on the need for spacecraft autonomy were time to criticality and the reduced situational awareness due to latency and reduced communication bandwidth. Time to criticality is related to time to effect; it is the time between first sign of a problem and that problem critically affecting operations in the absence of human intervention (crew or ground). Time to criticality can be lengthened by system design and the incorporation of autonomous functions. Data is key to spacecraft management; the ground controllers' timely access to data will shrink rapidly with distance from Earth. Ground control provides significant data analysis in current International Space Station (ISS) operations. In future deep space missions, more data collection, storage, prioritization, processing, and

analysis will need to happen without humans in the loop. As such, the constraint that will be placed on access to data drives much of the need for autonomy.

Several technology gaps based on the autonomy drivers and the dormancy functions required were found. Currently, the crew commonly provides sensing, sampling, and processing, and all ISS sensor data is delivered to the ground where nearly all data analysis happens. Contingency management across many subsystems, particularly leaks and emergencies (failures that currently require hands-on response from crew), currently relies heavily on both the crew and ground control. Data management and the associated situational awareness that data provides will have to be redesigned. An onboard vehicle system manager, capable of cross-vehicle state assessment, fault response, planning, and commanding, is therefore a major need.

Several recommendations are discussed in this study. Technology development needs based on the autonomy gap analysis are described. The most impactful technologies across all systems include sensor network design, system health management, planning, verification and validation of autonomous systems, and data analysis for situational awareness. Several trade studies are discussed, and three trades that affect the overall spacecraft architecture are recommended. The first trade centers around the role of robotics in the dormant spacecraft as mobile sensors. Data is hugely impactful to the autonomy of the spacecraft, and the collection of data is the first step. Due to the many types of telemetry required throughout the spacecraft, it may be more cost effective to make some types of sensors mobile. The other two trades have to do with self-actuation of valves and the sparing philosophy. Ubiquitous actuation may require more mass and a more complex strategy for redundancy than using robotic manipulators for certain interfaces. Likewise, in-place redundancy may be less complex than a common spares philosophy, but may require more mass. Investments in both the technology development areas and the system architecture trades are encouraged.

Many architectural requirements have been derived from this study. Requirements are given at the system level and the subsystem level, and there is a category of requirements that are data-driven. The most important requirement, however, is the need to determine early what the interconnections and interdependencies between the systems are. System design enables autonomy, and having tightly controlled interfaces between the subsystems is essential to reducing the complexity of the overall system, including the autonomous vehicle systems manager required to coordinate vehicle functions.

In summary, system design will play a major role in the autonomy of uninhabited spacecraft. Spacecraft must be designed with the need for autonomy in mind. The early definition of subsystem interdependencies is key. In addition to creating autonomous functions for vehicle systems management, strategies such as creating simplified interfaces, designing for less complexity, and selecting materials for more robust design are essential to increase the time to criticality for faults and failures in the dormant spacecraft. Currently, many contingency procedures are heavily reliant on crew. This reliance can be removed by smart designs, by implementing robotics solutions, or by increasing the risk posture of the mission to allow for suited repairs of the spacecraft upon the arrival of the crew. Autonomy will be essential for successful dormant operations of future deep space human spacecraft.

Table of Contents

1.0 Introduction

Various concepts of operations for human exploration of Mars center on the staged deployment of spacecraft, logistics, and crew. Though most studies focus on the needs for human occupation of the spacecraft and habitats, these resources will spend most of their lifetime unoccupied. As such, it is important to identify both the operational state that the unoccupied spacecraft or habitats will be in as well as what technology will be required to give the system the appropriate level of autonomy. This document attempts to address both of these things.

This study is separate from, but relies upon, previous and current work towards concept and requirement definitions of future exploration missions, such as the Evolvable Mars Campaign or the Future Capabilities Team. While these teams focus on developing the overall mission plan, resource needs, mass estimates, and requirements, the focus here has been identifying functions and technology gaps for the uncrewed mission phase. Likewise, this work follows from "Spacecraft Operations during Uncrewed Dormant Phases of Human Exploration Missions," a NASA white paper published in 2015. However, this work is more focused in that it centers around both a particular mission phase and the autonomous functions that will be needed to support the chosen mission.

The operational state of an unoccupied human spacecraft has been described as "dormant." This terminology will be used within this work; however, the connotation of "dormancy" may not aptly describe the operational state of these spacecraft, which may execute several functions while uninhabited, such as maintaining specific orbits, acting as a communications relay, or conducting science experiments. Though this work is based upon a Mars example, it will apply to uninhabited phases of the Deep Space Gateway (DSG) habitat as well. In particular, the intent of the DSG operations is to learn how to execute future exploration vehicles and missions of the future. As such, the plan is to limit communications to the uncrewed DSG to 8 hours per week, and to take an operations approach as close to the deep space robotic spacecraft as possible. The overall mission concept for the DSG is that it is a technology demonstrator for deep space missions, and all of the same operational phases described in this work will apply for the DSG missions as well. In short, this document can be used as a study of the autonomous systems technology needs for uncrewed phases that are essential for the DSG to fully accomplish its stated mission.

The definition of autonomy that will be used is as follows:

Autonomy is the state of existing or acting separately from others. (Merriam-Webster)

In this case, autonomy enables the dormant spacecraft to exist and act separately from crew and ground control. For this example mission, the dormant spacecraft will be need to be autonomous from ground control for 2-3 weeks during superior conjunction. More requirements for autonomy will be derived from latency, bandwidth, and resource availability constraints.

This definition of autonomy does not distinguish between autonomy that is local and reactive or autonomy that is proactive, planned, and coordinated. Though the types of technologies that are needed to achieve solutions that are simple local reactions versus more complex coordinated procedures are different, they will all fall into the bucket of autonomy for the intents and purposes of this study. The reason for this is simple. Both of these types of autonomous technologies (and more) will be required to make a dormant spacecraft in Mars orbit successfully meet its autonomy requirements.

The International Space Station (ISS) has many local autonomous reactions for faults and failures that have short time to effect in operation today. These technologies will not be the focus of this document, instead, an acknowledgement of their existence will be issued and the concentration will focus on the advances needed beyond the current operational state-of-the-art. Attempting to distinguish between qualities of autonomous systems would detract from the true focus of this document. However, the hardware, software, and avionics architectures that will be required to support the autonomous functionalities that are proposed herein will be discussed, as these architectures in themselves may require research and technology development.

In addition to providing a realistic assessment of what dormancy entails for human spacecraft and exploring gaps in the state-of-the-art for autonomous capabilities, recommendations for investments in autonomous system technology development and paths for technology infusion will be provided. The role of systems engineering in the creation of autonomous systems will be discussed. Finally, requirements for system design and architecture that are driven by the study's findings on autonomy for dormancy will be listed.

1.1 Process

The process by which this study was conducted is as follows. First, a design reference mission was determined, and mission parameters and assumptions were generated. A functional breakdown of the tasks required of each subsystem for several phases of the dormant period was then conducted. The four phases are defined as nominal operations, transition operations (entering dormancy from crewed operations and vice versa), contingency operations, and preventative maintenance and logistics. Additionally, the state-of-the-art in various tasks of the subsystem was determined, and compared against a desired value for the design reference mission in question. The gaps in ability were assessed and compared against the functional breakdowns to determine what autonomy needs must be developed. A development plan for these technology gaps was determined, and an assessment of the criticality and redundancy needs of this spacecraft and its subsystems with respect to similar subsystems on a robotic spacecraft was completed.

1.2 Outline

This document is organized in the following way. The next section, Section 2.0 describes the mission that was chosen along with the assumptions that were made. It includes the definitions and treatment of the autonomy drivers for each of the subsystems. Section 3.0 breaks down by subsystem or component the functions that are needed for the various phases of dormant operations. Each functional breakdown has a corresponding section on the autonomy needs based on the gaps shown in the autonomy drivers section, as well as a development plan section for the autonomy needs. Each subsystem also has a treatment on its criticality and expected redundancy. Section 4.0 is on the recommendations of the study team, and it includes thoughts on system engineering and system design requirements. The report then concludes with some thoughts towards future work.

2.0 Mission Parameters and Assumptions

The mission that was chosen for this design reference mission paper is based around a crewed mission to Mars. This study focuses on just a small part of this mission, in particular, it is assumed that the crew will travel to Mars in a spacecraft that will remain dormant in Martian orbit while the crew carries out a surface mission. This document will detail the functions that are needed to maintain the dormant spacecraft in a state such that it can be used for the crew's return trip to Earth.

Though the aim is to keep the scenario as broad as possible, several assumptions are made that will help frame the discussion on autonomy. First, it is assumed that the mission time at Mars is roughly 500 days; no assumptions are made about how long the humans are on the surface versus on board the spacecraft, however, it is assumed that the superior conjunction happens when the spacecraft is in the dormant state. A superior conjunction occurs when, from Mars, the Earth is on the opposite side of the sun. During this time (generally 2-3 weeks), little to no communications will be possible between assets and crew on or orbiting Mars and ground support on the Earth. This assumption provides a clear challenge for systems to be able to maintain the spacecraft with no earthbound ground support for a determinate time period.

Furthermore, it is assumed that the Earth will not be able to communicate with the spacecraft while it is on the opposite side of Mars from the Earth. This period is assumed to be 1 hour in duration, as a low orbit around Mars is about 2 hours long. The crew will have direct communication ability with the spacecraft, but this will be limited by ground passes and limited satellite-to-satellite relay capabilities in Martian orbit. A key assumption in regards to the crew is that they will have all commanding and telemetry capabilities that are available to the ground support on Earth. However, it is prudent to consider this ability to be used in critical or contingency scenarios only.

The spacecraft will likely have tasks supporting the crew on the ground, such as communication relaying to Earth, imaging (to support exploration or weather prediction), or communication inter-relay between Martian satellites. Other uses for the spacecraft (such as power generation or emergency logistics support) may be possible, but are not explicitly described in the functional breakdown that follows. It is assumed that flight software updates will be standard procedures.

In many cases, there are different technologies that are under development for use on an eventual Martian mission. An attempt to remain agnostic to technology types for as much of this document as practical is made. In the case that a choice between technologies is required for completeness, the top two to three potential technologies will be discussed.

2.1 Definition of Autonomy Drivers

The autonomy drivers are measures for each subsystem or component that will be used to generate autonomy needs. These drivers are system parameters and they have no assumptions for what type of autonomy is present or needed. These drivers, in conjunction with the mission parameters, set the interaction requirements for each subsystem. The difference between what the state-of-the-art currently is and what the requirements for the Martian mission described here is the crux of the analysis contained within. All autonomy needs discussed herein will be derived from a gap between what is needed and what exists currently; autonomy for the sake of technological progress will be avoided in this document.

Some initial definitions to ground the reader in basic terminology follow.

State-of-the-Art: This describes what the performance of systems is now, either on the ISS or on relevant ground analogs, if the ISS is not pertinent to that component.

Desired: This describes what the performance of systems will need to be for the design reference mission.

Safing: This is an action that can be taken by a system that has experienced a fault or failure in a subsystem to reduce or remove the functionality of that subsystem further in order to protect the health of critical subsystems of the vehicle.

There are four drivers that were deemed to be important to creating requirements for autonomy. They are defined as follows.

Mean Time Between Commands (MTBC): The MTBC is a measure of how often planned or expected unplanned commands must be given to the system. These planned commands may be due to seasonal or expected adjustments that must be made. The expected unplanned commands may be responses to faults or failures that are part of the operational (or "known") fault tree. The "commands" are human-driven (either ground or flight crew) and include not only the communications that effect a change in configuration of the flight system but also confirmations of automatic actions taken by the flight system.

Mean Time Between Interference (MTBI): The MTBI is a measure of how often unplanned human intervention is necessary. This includes unknown, unexpected off-nominal situations and differences between environment or system models and reality. The interference could be changes in software, configurations, or operations, and are currently human-driven.

Mean Time Between Telemetry (MTBT): The MTBT is a measure of how often ground control needs to receive data from the system in order to sufficiently monitor its state.

Time to Criticality (TTC): Time between the first sign of a problem to the problem critically affecting operations due to no human response. This time may be delayed by using safing techniques and other autonomous capabilities.

The four autonomy drivers above were determined based on operational needs for the system and the mission. Any mismatch between the state-of-the-art and the desired value of these drivers is the impetus for determining what types of autonomy will be needed to close these gaps. The design reference mission parameters that helped determine the values of these drivers include communication time delay, bandwidth, operational models, and environmental conditions such as the superior conjunction.

2.2 Autonomy Requirements

These autonomy requirements are derived from the mission parameters and provide guidance for how the desired values of the autonomy drivers for each subsystem were found.

1. The spacecraft must be autonomous from ground control for a contiguous 2-3 week period during its dormant phase.
Justification: Mars' superior conjunction with the Earth is for 2-3 weeks, and given the launch windows currently being considered, this is likely to happen during the dormant phase of a mission.

2. Each subsystem on the spacecraft must have a MTBT of over 1 hour.
Justification: Based on the spacecraft's orbit around Mars, there will be 1-hour periods of no telemetry.

3. Each subsystem on the spacecraft must have a TTC of several hours or more.
Justification: Based on the spacecraft's orbit around Mars, the crew on the ground would be able to react during superior conjunction in an emergency response when the spacecraft's ground track is amenable to communications. So, the TTC must be longer than the longest communication blackout between the Martian crew and the vehicle.

2.3 Subsystem Autonomy Drivers

In this section, an assessment of the state-of-the-art in each of the autonomy drivers defined above is given for each subsystem except the Vehicle Systems Manager (VSM), which does not have a clear operational example to pull from. Then, a desired value for each of the autonomy drivers is given. These values follow directly from the autonomy requirements given above, but also take into consideration an expert's assessment of where a subsystem's operation should feasibly be for this design reference mission, given the workload of ground control and the potential for technology advances.

For each of these subsystems, gaps have been found between how systems are currently operated and where they need to be for this design reference mission. In some cases, the gaps are large systems that are largely operated using crew and ground support now will need to operate (perhaps at some reduced state) during uncrewed periods. The data presented in this section compared against requirements for the Martian reference mission give unequivocal evidence that developments are needed to enable the proper amount of autonomy for the overall spacecraft. These developments will be discussed as part of the Dormancy Analysis in the next section.

2.3.1 Avionics

The avionics subsystem can be broken into four main tasks, shown in Table 1 below. The first task, "Maintain Current Avionics State," represents the nominal continuous operational state. Currently, commanding is rarely required to maintain this state, with weeks between commands. Also, core avionics relies on demonstrated high reliability and typically runs until failure. Failure times are often on the order of months or years. As such, there is no gap in autonomy for this task with respect to nominal or unexpected off-nominal commanding. However, communication of state data to maintain the current avionics states is monitored a more frequent rate. Some higher activity periods such as rendezvous and docking are expected to entail nearly constant monitoring by either the ground or crew. While some operational controls may be present to mitigate this requirement for the uninhabited phase, even the nominal system state has typically been monitored from the ground, with ground systems capable of alerting controllers when there is a change detected in the system. For short loss of communication (LOC) periods, failures could be addressed by autonomous Fault Detection, Isolation, and Recovery (FDIR). Upon acquisition of signal, the ground could work toward obtaining a better, more desired state. Currently, getting to a better, more desired state after a failure generally results in one of the following actions, usually taken in order. First, ground controllers collect detailed information/logs from the affected box and look for any 'smoking gun' to determine whether the cause of the failure is knowable and if it is, determine if the failure is likely to be permanent. Then the box is power cycled to see if it recovers. If it does not recover, the crew is instructed to exchange it for another box or exchange the affected component within the box. Expected significantly longer loss of communication scenarios imply that the "obtaining a better, more desired state" needs to be addressed onboard.

The second task focuses on times when changing the current avionics state is required. In particular, nominal commands would be to bring up or take down avionics hardware in support of subsystem desired changes. This would largely be for configuration changes associated with tasks in support of higher intensity operations such as rendezvous and docking. As such, the times that this task would be necessary during dormant operations may be reduced, but this activity is still expected in order to transition from one desired avionics state to another. It is expected that this would largely occur during periods in which the desired end state could be verified by the ground, but a state transition could also occur in a time initiated transition via a series of stored command sequences. A failure could invoke a non-desired or incomplete state transition. For instances in which contact with the ground is available, a holding pattern based upon an accepted interim state would need to be built into part of the transition scripting. For extended periods without communication, either the state transition needs to be precluded or autonomy needs to be in place to either return to the old known state or press forward through alternative means to the new state. Telemetry in support of state transition will be necessary in order to determine the successful execution of steps associated with the transition as well as verification that the final transition has been successful. Because of significant time delays, critical transition activities will require some degree of autonomy, somewhat dependent upon the pace of desired transition in conjunction with time latency. Time to criticality will be operation dependent. In general, for operations with long timelines, a fallback position of going back to the initial state may be acceptable until the issue can be remedied by the ground. For short duration operations that must occur, workarounds/autonomy to get to the new state would be necessary.

Table 1: Avionics Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Maintain Current Avionics State	SOA:	Weeks	Months	Minutes	Minutes
	Desired:	Weeks	Months	Weeks	Months
Change Avionics State	SOA:	Minutes	Months	Seconds	Minutes
	Desired:	Minutes	Months	Days	Months
FDIR (Immediate Time to Effect)	SOA:	Weeks	Minutes	Seconds	Seconds
	Desired:	Month(s)	Months	Weeks	Months
Predictive/ Prescriptive FDIR (Long Term Time to Effect)	SOA:	N/A	N/A	N/A	N/A
	Desired:	Months	Months	Days to Months	Months

The third task involves response to faults immediately after they occur. Nominally, component failures are expected at a certain rate. Because of resupply and latency concerns, higher reliability is going to be needed to reduce the rate that FDIR occurs. This can be addressed through internal design redundancy and creative degraded mode operations. Unexpected Built-In Test (BIT) signatures and responses are examples of problems that may not have specific recovery procedures, and long communication outage periods will imply that FDIR, particularly the isolation and recovery aspects, will need to be addressed by autonomy where they historically could have been handled by ground intervention. Telemetry is required for the diagnosing and analysis of failures, and as such the ground will need to understand any changes to the last know configuration, including any onboard autonomous FDIR actions during the LOC time period. The time to criticality will be architecture dependent and will depend upon the ability of the overall architecture to adjust and adapt. Because avionics supports most other subsystems, there is a risk that a component failure in an avionics box could trigger an undesirable response in other systems; for example, a sensor signal that results from an avionics failure could trigger an emergency response that may leave the overall spacecraft in an off-nominal state. In this case, the time to criticality could be artificially low due to the lack of coordination between subsystems. The ability to address this will be dependent upon backup capabilities as well as hardware and software distribution within the architecture.

The fourth task is based on a prognosis ability that does not currently exist on orbit. For avionics, this is defined as an ability to create revised or alternative remedies based upon avionics trending information, including degraded mode operations. Currently, ground controllers monitor for trending behavior. Due to extended LOC periods, trending may need to occur onboard with adjustments to what is being monitored based upon changes to the expected system state. This ability could help adjust both nominal and expected fault responses as well as tune strategies for dealing with the unexpected. The need for telemetry for updates to the system should be infrequent since this is data for trending. With appropriate architecture and FDIR, the time to criticality is expected to be long for preventative trending needs.

2.3.2 Communications

The communications subsystem autonomy drivers are shown in Table 2. Typical commanding includes switching antennas due to high beta angle[1] or changing data rates or frequencies due to operational needs, degraded conditions, or signal interference. These commands are already at an acceptable frequency for this reference mission. Likewise, commanding to recover from an unexpected anomaly is currently done only rarely. Usually the communications system's FDIR algorithms takes it to a known state if it detects loss of communications with Earth/other interface or other anomalies. When the spacecraft is crewed, the communications system's status is checked periodically. However, checking the status of the communication system during quiescent operations will need to happen less frequently. Likewise, since communications failure criticality depends strongly on the system's response to no communications given its state and the current mission phase, autonomous recovery may be required to reduce the time to criticality for this reference mission.

Table 2: Communications Autonomy Drivers

	MTBC	**MTBI**	**MTBT**	**TTC**
SOA:	Months	Months	Days	Minutes* to hours
Desired:	Months	Months	Weeks	Days

**Depending on when the problem occurs, it could be minutes to hours/days before it could critically affect the operations.*

2.3.3 Environmental Control and Life Support System

The Environmental Control and Life Support System (ECLSS) performs four key tasks, which are described in Table 3 below. The first task, atmosphere conditioning, is to ensure the crew has breathable air and a livable temperature. This includes the removal of contaminants, control of humidity, and temperature control. The ECLS performs this task during the uncrewed dormant phase in order to mitigate damage to other systems, though the control bands (i.e., temperature, humidity) may be wider than during the crewed phases. A minimum temperature would be established to mitigate freezing of fluid systems, and humidity control is needed to mitigate the formation of free liquid due to condensation of ambient moisture. An example of nominal commanding is stopping humidity removal several days into the dormant period (if necessary), and an example of unexpected commanding could

[1] https://en.wikipedia.org/wiki/Beta_angle

be to adjust heater or fan settings when off-nominal temperatures or thermal distribution is detected. Because a habitable environment is not necessary during dormant operations, long response times to off-nominal parameters are acceptable and periodic status checks for phase planning is sufficient. However, since temperature and humidity affecting other subsystems' performance are credible failures, autonomous functions to increase the time to criticality are important.

The second task is pressure management, ensuring that sufficient total pressure and oxygen are present to support the crew during crewed operations. The vehicle/ECLSS controller may control to a lower total pressure during dormancy in order to conserve consumables. The pressure control band will be driven by systems in the pressurized volume. The oxygen supply system is expected to be deactivated during dormancy in order reduce the flammability hazards. It is a goal to have a robust FDIR capability to resolve issues at the vehicle and ECLSS controllers. An overpressure situation could compromise the habitat structure; an under-pressure situation could compromise other vehicle systems in the pressurized volume required for a safe return trip. As such, the time to criticality without human intervention must be improved for this reference mission.

Table 3: ECLSS Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Atmosphere Conditioning	SOA:	Minutes	Days	Minutes	Minutes (high ppO2) to Months
	Desired:	Days to Months	Weeks to Months	Weeks to Months	Hours to Months
Pressure Management	SOA:	Minutes	Days to Weeks	Seconds to Minutes	Minutes to Hours
	Desired:	Weeks to Months	Weeks to Months	Weeks to Months	Hours
Water Management	SOA:	Hours	Days	Minutes	Minutes (leak) to Weeks (quality)
	Desired:	Weeks to Months	Weeks to Months	Months	Hours (leak) to Months (quality)
Environmental Monitoring	SOA:	Minutes	Minutes	Minutes	Dependent on the system being monitored
	Desired:	Days to Months	Weeks to Months	Weeks to Months	Hours to Months

The third task is water management. Provision of potable water is necessary to support crew life. This is not required during dormancy but actions will be needed to protect the systems so that it is functional when the crew returns. Since potable water is not required during dormant periods, the water recovery system is expected to be off. Some commanding may be needed if active biocide or water cycling is implemented. Systems may also be commanded if off-nominal parameters are observed. The time to criticality depends on the type of failure that might occur. A leak, for example, may result in damage to other spacecraft systems but the problem of contaminated water may be resolved when crew returns. Automation of water system fault detection should be improved to increase the time to criticality without human intervention.

The fourth and final task, environmental monitoring, will provide an indication of dangerous conditions and is necessary to determine whether the atmosphere is acceptable upon crew return. Since a breathable environment is not necessary during dormant operations, long response times to off-nominal parameters are acceptable and periodic status checks for phase planning are sufficient. Nominal commanding for this task includes requests for data, and an unexpected commanding example could be running calibrations. Catastrophic failures associated with elevated oxygen and total pressure management due to component failures are credible during dormancy and so autonomous functions to increase the time to criticality without human intervention are important.

2.3.4 Guidance, Navigation, and Control

The guidance, navigation, and control (GNC) subsystem has three main tasks during dormancy for this reference mission, shown in Table 4 below. The first task is attitude control, the maintenance of a commanded rotational state. Attitude control requirements include thermal control of the vehicle, maintenance of communication, and state control for proximity operations. Onboard sensors, such as star trackers, exist to provide the vehicle with rotational state. Currently, the ground requires high-rate telemetry for safety. In the future, the onboard system must manage onboard attitude commanding when communication lags prevent responsive change from ground control, or during extended loss-of-communication events. Errors that result in unexpected changes in attitude or rates would need to be dealt with quickly. Autonomous error management is required to improve response time in order to reduce the time to criticality without human response.

Table 4: GNC Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Attitude Control	SOA:	Hours to Days	Months to Years	Minutes to Hours	Seconds to Minutes
	Desired:	Months	Years	Hours to Days	Hours
Trajectory Control	SOA:	Hours to Days	Days to Months	Minutes to Hours	Seconds to Minutes
	Desired:	Months	Months	Hours	Hours
Navigation	SOA:	Hours	Months to Years	Minutes to Hours	Minutes to Hours
	Desired:	Months	Years	Hours	Hours

The second task is trajectory control, the maintenance of translational dynamic state to support mission phase. This task includes ascent trajectory, orbital position maintenance, transfer burns between orbital bodies, and entry at the destination or Earth. In nominal operations, onboard trajectory planning takes into account vehicle state and consumables and supports burn plans. In some cases, abort planning must be performed to take into account unexpected failures. This abort capability is limited based on mission phase and vehicle capability. For instance, if the vehicle is en route to Mars, the types of abort trajectories are limited. Ground requires high-rate telemetry state for crew safety, so an increase in onboard trajectory commanding is required when communication lags prevent responsive change. Errors that result in unexpected changes in position or rates would need to be corrected quickly. Autonomous error management is required to improve response time in order to increase the time to criticality without human response.

The third task is navigation, the determination of vehicle state based on onboard sensors. To date, onboard sensors have a limited ability to manage error growth, requiring occasional "navigation updates" from the ground. The goal for future spaceflight is to identify onboard sensors capable of limiting navigation error growth and reducing the frequency of ground updates. An example of expected commanding is scheduled ground updates required to correct onboard navigation error growth. One way to reduce this commanding is by allowing onboard navigation sensors to manage this error growth. For cases of sensor failure where unexpected updates are required, some capabilities include on board sighting systems (like SEXTANT) to help with attitude or with position (if a body is nearby). As with the other tasks, the ground requires high-rate telemetry state for safety, so an increase onboard attitude commanding is required when communication lags prevent responsive change. The key for navigation state is to understand the vehicle dynamics well enough to make mission trajectory decisions. These decisions could include mid-course corrections, acceleration commands (for constant-thrust propulsion), and for mission aborts if possible. Onboard navigation sensors must preserve state error below a threshold where these trajectory commands can be executed successfully. Otherwise, the commanded burns will not achieve the correct final state and the mission could be jeopardized.

2.3.5 Power

The power system is decomposed into three tasks, shown in Table 5. The first task is power generation, which includes hardware such as the solar arrays. Nominal commanding in this task includes commanded changes to set points (i.e., nominal battery state of charge ranges). Unexpected commanding could involve changing operating conditions or set points to manage reductions of capabilities in contingency situations. This currently happens at a sufficiently low frequency to satisfy the requirements of this reference mission. The need for telemetry on the ground could be reduced by creating autonomy capable of recording high speed snapshots specifically of significant events, and then prioritizing these snapshots for downlink. The time to criticality of a power generation failure depends on ride-through capabilities of energy storage.

The second task is energy storage, which includes hardware such as the batteries, fuel cells, or electrolyzer. Nominal commanding similarly includes reconfiguration and set point changes; these could be commanded based on local conditions rather than external commands from the ground. Likewise, unexpected commanding could involve changing operating conditions or set points to manage reductions of capabilities in contingency situations. The need for telemetry on the ground could be reduced by creating autonomy capable of recording high speed snapshots of significant events. Time to criticality could be due to a range of events from equipment failure to unanticipated wear-out, and autonomy is needed to move the need for human intervention out to an appropriate time frame for the reference mission. For example, if the Power Management and Distribution (PMAD) system could autonomously load shed to dynamically respond to reduced energy storage and could autonomously connect alternate energy storage sources (cross connect power busses), then the time to criticality could be increased.

The third task is power management and distribution. Nominal commanding could be due to preplanned reconfigurations as part of standard mission sequences, and unexpected commanding could be external commands to optimize system state after rapid automatic reconfiguration in response to faults like short circuits. The need for telemetry on the ground could be reduced by creating autonomy capable of recording high speed snapshots of significant events. Time to criticality could be due to load faults or equipment failure, and autonomy is needed to move the need for human intervention out to an appropriate time for the reference mission. For example, if the PMAD system could sense and respond to anomalous load currents (arc faults, series arcing from loose connections, ground fault currents) before they progressed to short circuits, failure propagation could be reduced, and TTC lengthened.

Table 5: Power Management and Distribution Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Power generation	SOA:	Day to Week	Months to Years	Seconds to Minutes	Hours to Days
	Desired:	Months	Months to Years	Hours to Days	Hours to Days
Energy storage	SOA:	Months	Months to Years	Seconds to Minutes	Milliseconds to Years
	Desired:	Months to Years	Years	Hours to Days	Hours to Years
Power Management and Distribution	SOA:	Months	Months	Seconds to Minutes	Milliseconds
	Desired:	Months to Years	Months to Years	Hours to Days	Hours to Years

2.3.6 Propulsion

The propulsion system can be decomposed into six tasks, shown in Table 6. The first two tasks involve providing thrust for attitude and translation commanding. Thrusters are fired autonomously per the jet table onboard and fault flags. Sometimes ground ops may have other jet preference for life, performance, other reasons, for both nominal and unexpected commanding. Telemetry is required to track jet usage, and jet failures can quickly cause a critical state depending on the current operation.

The third task is configuring the valves for pressurization of the tanks. Examples of nominal commanding include opening the valves prior to a translation burn, closing pressurization valves after translation burns, or operating the attitude control system (ACS) in blowdown with all pressurization valves closed. Pressurization valve or regulator leakage may require these to be kept closed to prevent over pressure of the propellant tanks and relief device operation. A pressurization valve failure in the closed position would result in off-nominal configurations such as using a cross-feed system. An autonomous function is required for checking the valve states and monitoring for valve leakage or full open failures. These would result in burst disc rupture and relief valve operation that require immediate response to avoid the catastrophic loss of helium.

The fourth task is determining the propellant storage temperature, to avoid propellant freezing or maximum temperature exceedance. Nominal commanding may be used to adjust the set points of propulsion thermal control. Unexpected situations such as heater failures may require the selection of different strings or a change in spacecraft attitude. Telemetry is required to monitor for failures; failure to keep the propellant in the proper state could cause a critical situation.

The fifth and sixth tasks involve assessing the quantity of propellant and pressurant gases remaining and detecting leakages in tanks, valves, or engines. Nominal commands are used for planning burns and assessing health of tanks, lines, and engines. In extreme cases, insufficient propellant may require changes to mission. Quantity gauging may involve looking at telemetry and performing ground calculation. Quantity gauging plus other leak detection methods may require rapid response to isolate the component or system if a significant quantity decay or leak is detected.

Table 6: Propulsion Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Provide attitude thrust	SOA:	Hours to Days	Hours to Days	Hours to days	Seconds
	Desired:	Days	Days	Days	Hours
Provide translation thrust	SOA:	Hours to Days	Hours to Days	Minutes to Hours	Seconds
	Desired:	Days	Days	Hours to Days	Hours
Valve Configuration for pressurization	SOA:	Hours to Days	Hours to Days	Minutes to Hours	Seconds
	Desired:	Days	Days	Days	Hours
Propellant storage temperature	SOA:	Hours to Days	Hours to Days	Minutes to Hours	Hours
	Desired:	Days	Days	Days	Days
Provide qty of pressurant and propellant remaining	SOA:	Hours to Days	Hours to Days	Minutes to Hours	Seconds
	Desired:	Days	Days	Days	Hours
Detect tank, valve, or engine leakage	SOA:	Hours to Days	Hours to Days	Minutes to Hours	Seconds
	Desired:	Days	Days	Days	Hours

2.3.7 Robotics

The International Space Station has two robots used for servicing, the Space Station Remote Manipulator System (SSRMS) and the Special Purpose Dexterous Manipulator (DEXTRE), as shown in Figure 1. The Mobile Base System is a system that transports these robots to various points around the ISS. The operational model for these robots involve constant human control and supervision, even if a pre-programmed set of joint moves is being performed. No robotic operations are completed without crew or ground controllers involved, and no technology beyond basic joint teleoperation has been used for controlling these manipulators. Research has been on going, however, on ground systems that improve upon the capabilities of the SSRMS and DEXTRE for the functions of mobility, manipulation, and inspection. In the derivation of the autonomy drivers for robotics, a combination of the operational ISS robots and selected ground robots were used to determine the drivers. In particular, the Robonaut 2 climbing legs are used as an example of intravehicular mobility, whereas the Robonaut 2 arms are used as examples of manipulation. These are depicted in Figure 2. In particular, these are used as models for the state-of-the-art in MTBI, due to the assumption that the more dexterous manipulation and smaller scale mobility will be required for the dormant spacecraft. However, MTBC, MTBT, and TTC are modeled after the ISS operational robots. Though the ISS robots are used for inspection, the Astrobee robot, shown in Figure 3, is used instead in all numbers except the TTC, for which requests for visual inspection by crew members on the ISS is used. In all cases, all numbers are taken as a mean over the time that the robots are operating (i.e., time when the robots are off or stowed does not contribute to the mean).

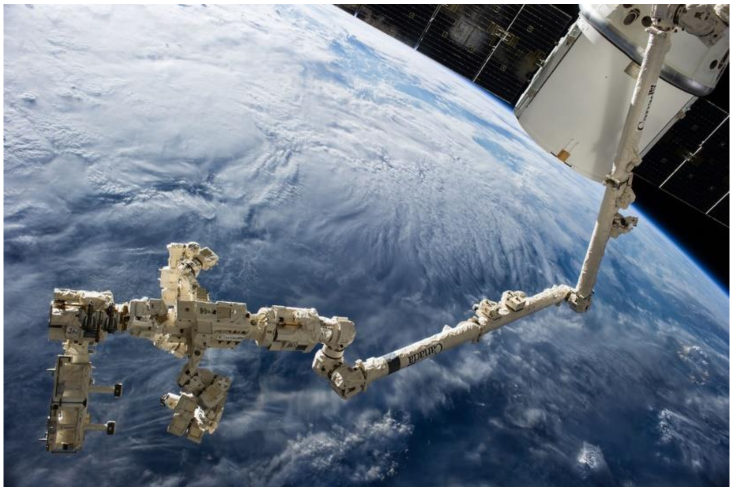


Figure 1: SSRMS and DEXTRE

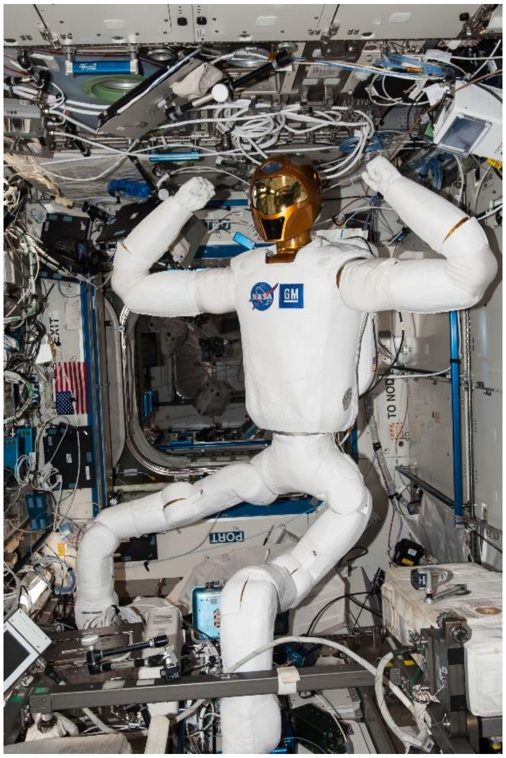


Figure 2: Robonaut 2

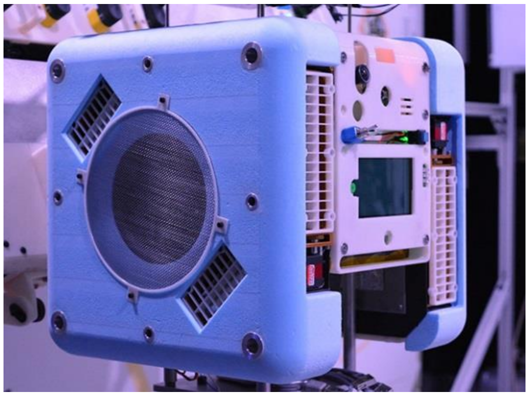


Figure 3: Astrobee

The robotics subsystem can be decomposed into three tasks, listed in Table 7. The first task is mobility, which involves moving assets from one point to another. Nominal commanding includes human or system commands on an activity level, such as adjusting for plans not found or unsuitable configurations. Unexpected commanding could be recovering from faults in the system. Telemetry is needed in order to see the status of commanded tasks or fault states. Mobility platform failures could become critical if positioning a manipulator to complete a robotic task (i.e., maintenance, capturing a logistics module) becomes critical.

The second task is manipulation, which is required when a repair must be made or other maintenance or logistics task must be performed. Nominal commanding includes human or system commands on activity level, such as help with tool placement or identification. Unexpected commanding could be recovering from faults in the system. Telemetry is needed in order to see the status of commanded tasks or fault states. Manipulators could become critical if actions such as capturing logistics module or recovering from a failure become critical.

Table 7: Robotics Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Mobility Platforms	SOA:	Minutes	Days	Minutes	Days to Weeks
	Desired:	Days	Weeks to Months	Day	Days to Weeks
Manipulators	SOA:	Minutes	Days	Seconds to Minutes	Day
	Desired:	Days	Weeks to Months	Day	Day
Sensing & Perception (Inspection)	SOA:	Tens of minutes	Hours	Seconds to Minutes	Hours to Weeks
	Desired:	Days	Weeks to Months	Day	Hours to Weeks

The third task is sensing and perception for inspection. Nominal commanding includes human or system commands for an inspection routine. Unexpected commanding could involve resetting faulted hardware, and telemetry is required for acquiring the inspection data. Inspection robots could become critical if the need to inspect a failed system becomes critical.

2.3.8 Software

The software subsystem can be decomposed into seven tasks, shown in Table 8. The first task maintains the current spacecraft state. Some nominal commanding is expected to do this, and further commanding is required for unexpected state changes, such as rates or trends out of bounds. Telemetry is required to monitor spacecraft state, since critical situations can occur upon sudden or unexpected changes in spacecraft flight state.

The second task is active when changing the spacecraft's state, for example, altering the flight path or orientation, or transitioning to crewed flight. Unexpected commanding may be required if failures cause interventions that are needed to complete the spacecraft's transition. Telemetry is required to manage spacecraft state, since critical situations can occur upon sudden or unexpected changes in spacecraft flight state.

The third and fourth tasks involve fault detection, isolation, and recovery on various time scales. Immediate FDIR currently requires human commanding to resolve known or unexpected spacecraft faults. Telemetry is required for diagnostics and analysis, and criticality depends on the type of fault that occurs. Prognostic FDIR does not currently exist in the state of the art, but would deal with implementing remedies of common problems over a longer time horizon. Commanding might be required to revise or use alternative remedies, or even to alter a remedy when the unexpected occurs. Telemetry is required to identify and address subsystem trends, and faults requiring servicing or workarounds utilizing available redundancy would drive the time to criticality.

The fifth task is to recover to a degraded spacecraft state upon some fault or failure. Expected commanding in this situation may include subsystem corrective actions needed to restore lost capability or recover assets from a safe state. In special cases, unique fault resolution actions to recover lost capability or assets may be required. Telemetry is needed for extended ground diagnostics and analysis, and criticality is driven by faults requiring immediate attention to preempt further spacecraft degradation.

Table 8: Software Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Maintain Current S/C State	SOA:	Hours	Days	Seconds	Seconds
	Desired:	Months	Months	Daily	Months
Change S/C State	SOA:	Minutes	Daily	Seconds	Seconds
	Desired:	Months	Months	Days	Months
FDIR (Immediate Time to Effect)	SOA:	Days	Days	Seconds	Seconds
	Desired:	Month(s)	Months	Hours	Months
Prognostic FDIR (Long Time to Effect)	SOA:	N/A	N/A	N/A	N/A
	Desired:	Months	Months	Days/Month	Months
Recover to a Degraded S/C State	SOA:	Minutes	Minutes	Seconds to Minutes	Minutes
	Desired:	Month(s)	Months	Hours	Weeks
Scheduled Software Update(s)	SOA:	Weeks	Days	Minutes	Minutes
	Desired:	Months	Months	Hours to Days	Months
Immediate Software Update(s)	SOA:	Minutes	Minute(s)	Seconds	Minutes
	Desired:	Month	Month	Hours to Days	Weeks

The sixth and seventh tasks deal with software upgrades on scheduled or immediate time frames. Expected commanding may involve adjustments to calibration data and code enhancements, or even code patches for immediate upgrades. Unexpected commanding could be resolution of issues in the updated software capabilities or in the installation itself. For example, the software system would be able to determine whether the new update is faulty by analyzing fault types and trends. If that is the case, the software system would transition back to the prior version, providing information as to why. Telemetry is required for confirmation of proper application or diagnosis of issues, and criticality is driven by faults or bugs in the updated software or induced in existing software by the update that require immediate attention.

2.3.9 Spacecraft Emergency Response

The spacecraft emergency response system has been decomposed into three tasks, shown in Table 9. The first task is fire suppression. Nominal commanding currently consists of weekly commanding to detectors and Portable Emergency Provisions System inspection at approximately 40-day intervals on the ISS. An improvement on the current state is to automate these checks and inspections such that readiness can be continuously confirmed via telemetry. Unexpected commanding currently consists of resetting spurious alarms. False alarms are usually initiated by crew activity in a module, which can spread dust causing smoke detector obscuration sensors to register a problem. In dormant phases of flight with limited ventilation active, false alarms should not occur. Note that the current fire detection system on ISS is not a hazard control in that the fire hazard is controlled via system design and material selection, not by the alarms and responses in the detection system. System design for emergency response must require effective operation without unexpected commanding during all expected conditions and scenarios. Expected commanding is included to confirm readiness of the system to perform when needed. Telemetry is required to understand the state of the system and if any fires have been detected. Currently, fire emergency requires immediate response by the crew on the ISS. For the dormant mission, there must be some autonomous response or autonomy built into the system design that increases the time to criticality to hours.

Table 9: Spacecraft Emergency Response Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Fire Suppression	SOA:	Weekly	N/A	Seconds	Seconds
	Desired:	Months	N/A	Hours	Hours
Toxic Atmosphere Detection and Scrubbing	SOA:	N/A	N/A	Seconds	Seconds
	Desired:	Months	N/A	Hours	Hours
Hull Breach, Rapid Depress Detection and Isolation	SOA:	N/A	N/A	Seconds	Seconds
	Desired:	Months	N/A	Hours	Hours

The second task is toxic atmosphere detection and scrubbing. The criticality of a toxic contamination event during dormancy phases depends on whether vehicle systems are sensitive to the contaminant. If the vehicle systems are designed in a way that failures can cause contamination that can impact systems required for continued dormant operations, toxic contamination must be scrubbed from the atmosphere during dormant phases to mitigate the risk of those impacts. For contamination that is harmful to crew members but not harmful to vehicle systems (e.g. high CO_2 levels) the time to criticality is far longer. Systems must be in place to scrub the atmosphere of contaminants prior to the return of the crew. Nominal commanding is expected to include inspections of sensor systems and scrubbers. Ideally, the system should be designed so that it has no ability to release contaminants into the atmosphere. Currently, emergency toxic events fall into two categories: toxic spill and ammonia (NH_3) release into the cabin via a thermal system failure. Recognition of toxic spill events on the ISS relies on crew detection, and these events are not likely to be applicable to dormancy phases. Ideally, the vehicle design will preclude the possibility of a toxic substance release into the crew cabin due to system failure such that automated toxic release protection will not be necessary. Where such a design does exist, sensors within the affected systems will provide insight into the integrity of the system and any malfunctions. The current example of this situation is the Thermal Control System (TCS) on ISS, in which fluid sensors are used to monitor for a heat exchanger breach that would trigger an NH_3 release into the crew cabin. Telemetry is currently required for leak detection and response, and there are several limitations on thermal system configuration to prevent the possibility of a heat exchanger breach. For the dormant mission, there must be some autonomous response or autonomy built into the system design that increases the time to criticality of such an event to hours.

The third task is hull breach and rapid depressurization detection and isolation. Nominal commanding is expected to include inspections of sensor systems. Currently, rapid depressurization cases on the ISS depend on module pressure sensors and/or crew detection. Telemetry is currently required for leak detection and response. Time to criticality when no crew is on board is driven by the ability for onboard vehicle equipment, such as Command and Data Handling (C&DH) and Avionics, to continue to function as cabin pressure decreases to vacuum, and the potential need to isolate the leak to preserve atmosphere resources onboard the spacecraft. The response to a detected leak or depressurization would be to isolate the problem, identify the leak source, repair the breach and repressurize affected portions of the vehicle. For the dormant mission, there must be some autonomous response or autonomy built into the system design that increases the time to criticality to hours.

2.3.10 Structures

The structures subsystem has a single task, shown in Table 10, Structural Health Monitoring (SHM). SHM requires storage of all sensor data and periodic download and subsequent overwrite of all sensor data files.   Periodic download scheduling will be based on data storage allocation to SHM. Data exceeding predetermined limit values requires automatic download of the data from the sensors indicating load exceedances.   Additional data download may be required on demand for assessment of a load exceedance alarm event. Nominal commanding includes data collection and inspection of stored FDIR data for the system. Telemetry is required for all stored sensor data files and stored FDIR files. Failures of SHM capability will result in total lack of awareness about structural failures events, that may have occurred during the dormancy period. The time to criticality is not applicable for this task because the spacecraft never becomes critical due to a SHM failure.

Table 10: Structures Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Structural Health Monitoring	SOA:	Months	Months	Daily	N/A
	Desired:	Months	Years	Weeks	N/A

2.3.11 Thermal

The thermal subsystem has been decomposed into three tasks required for dormant spacecraft, shown in Table 11. The first task is the adjustment of thermal transport loop fluids, since the system is dependent on this (e.g., mixture, loop volume, flow path, flow rate). The standard capability of space vehicle thermal systems includes commanding or adjustments dependent on operations, environment changes, etc. Nominal commanding, therefore, includes preplanned checks and adjustments if needed as part of standard mission sequences. Adjustments due to off-nominal conditions or failures (e.g., leakage, pump failure, excess boil off) are possible unexpected commanding opportunities. Telemetry is currently required for sensor data and fault notifications. For nominal adjustments, the time to criticality before human intervention is required is days to months, however extreme cases occur when there is a significant critical failure, such as a leakage.

Table 11: Thermal Autonomy Drivers

Task		MTBC	MTBI	MTBT	TTC
Adjustment of thermal transport loop fluid(s)	SOA:	Minutes	Minute(s) to Hour(s)	Minutes	Minutes to Hours
	Desired:	Month	Month(s) to Year(s)	Hour(s)	Hours*
Off-nominal adjustment for heat rejection turn down	SOA:	Minutes	Minutes to Hours	Seconds to Minutes	Minutes to Hours
	Desired:	Daily	Months to Years	Hours	Hours*
Radiator Ops	SOA:	Tens of minutes to Hours	Tens of minutes to Hours	Minutes	Minutes to Hours
	Desired:	Hours to Days	Years to Decades	Minutes to Hours	Hours*

**Autonomy will be needed to increase this time to criticality for human intervention for this mission.*

The second task is off-nominal adjustment for heat rejection. Should a failure or off-nominal condition occur affecting power levels or the required heat rejection in the vehicle or module, adjustments may be required. Nominal commanding is assumed to be controlled via sensors and monitoring with automated adjustments but regular checks are needed for health and maintenance (i.e., load shedding). Unexpected commanding may involve adjustments such as changing operating conditions or set points (e.g., adjusting software parameters, opening/closing valves). Telemetry is currently required for sensor data and fault notifications and time to criticality is based largely on the need to preclude significant consequences (e.g., freezing).

The third task is radiator operations, as it may be advantageous to stow or deploy radiators for various missions or phases of missions assuming the same architecture or vehicle is used for multiple missions (e.g., Cislunar and Mars). Nominal commanding includes deployment and/or stow based on mission period. Unexpected mitigations may require extravehicular activity (EVA) for deploy or stow failures, or reconfiguration of the flow path if the radiator is non-operational. Telemetry is required for flow rates and other data. For deploy or stow failures, criticality depends on mission phase.

3.0 Component Dormancy Analysis

This section contains the functional breakdown of each subsystem or component (such as the Vehicle Systems Monitor or Robotics) during the uncrewed, dormant phase. One of the goals of this document is to provide a concept of operations of the uncrewed spacecraft. These component analysis sections attempt to do just that by discussing what functions will be required during different phases of the dormant mission period. The nominal operations phase includes all healthy, non-transition times. This is the status quo of the dormant period. These functional breakdowns will list what parts of the subsystem are operating and if operation is reduced in any way. The transition operations phases include two time periods. The first time period is directly after the crew has exited the spacecraft and is when the completion of the transition to nominal dormant operations is executed. The second time period is just before crew returns to the spacecraft, when the transition to crewed operations occurs. These sections list what actions are required after the crew leaves or before the crew returns, and so are technically still part of the uncrewed phase. Contingency operations sections discuss what must happen when various contingencies occur. These functional breakdowns are often the most challenging since many types of contingencies are currently handled by the ground controllers and/or crew members. These functional breakdowns also include a bit of a discussion on system criticality and redundancy. Finally, preventative maintenance and logistics support of the subsystem is handled separately.

The dormancy analysis continues for each component with a discussion of the gaps found in the autonomy drivers analysis with respect to the functions that are required during dormancy. The combination of the gaps and functions generate clear autonomy needs for each component. These are described in the Autonomous Functions Needed subsections. The analysis continues with the Development Plan subsections, which give a technology breakdown of the needs described. These development plans discuss technologies that are common to many components as well as technologies that are focused on just a few components. An assessment of the investment in these technologies outside of NASA is briefly given, where possible, to help direct funding choices in the future. Development plans are also broken down into several types of technologies, from hardware and systems engineering advances to software and algorithmic developments needed.

Finally, the criticality and redundancy subsections in most of the component analyses are given in order to distinguish the needs of an uncrewed human spacecraft from robotic spacecraft that have come before. These sections attempt to describe the system-level considerations of architecture that will be required in order to design the system for autonomy. A further treatment of the differences between robotic and human spacecraft is given in Appendix D, and more discussion of architectural requirements imposed by the need for autonomy can be found in Section 4.0.

3.1 Avionics

Avionics, in conjunction with software, can be thought of as an enabler of other desired vehicle subsystem behavior. This desired behavior can be established by a number of methods. System monitoring enables either the crew, ground via a communication link, or software to understand the subsystem state, maintain the subsystem state, or if so desired, to transition the subsystem state to a new state. This would include avionics' own subsystem monitoring capability. New subsystem state behavior is transitioned from the previous state behavior either intentionally or unintentionally. Intentional subsystem state behavior is initiated via commanding, either by a single command, a series of commands, or by initiation of a stored command sequence. Unintentional subsystem behavior is generally the result of a failure or erroneous commands placing the subsystem into the unintentional behavior. Failures can be addressed through appropriate system monitoring that allows detection as well as prior design considerations that allow for the isolation and recovery back to the desired subsystem behavior.

Some failures have short time to criticality (TTC), whereas others have longer TTC from failure initiation. Historically for crewed vehicles, critical subsystems with short TTC have often utilized software to enable autonomous fault detection, isolation, and recovery (FDIR). These were often mission phase specific and addressed during critical flight operations, such as during ascent or entry when the TTC could be so short as to be catastrophic. These mission phases precluded sufficient crew reaction time to address the failure, that is, its detection, isolation, and recovery before the critical failure took effect. Many failures on orbit, on the other hand, lack the low TTC of failures during dynamic phases such as ascent and entry. This allows sufficient time, once detected, for either ground or crew intervention. Because of low communication latency as well as limited expected loss of communication (typically less than 10 minutes for the Tracking and Data Relay Satellite System (TDRSS) Zones of Exclusion) in low Earth orbit, those failures that could be addressed by the ground or crew were generally not part of an autonomous FDIR software effort. For unexpected communication outages, the crew could be utilized to help reestablish communication. The use of the crew or ground for isolation and recovery, as opposed to autonomous FDIR, was usually the result of design drivers to reduce software complexity and cost and reduce the avionics utilization of a more complex software system, with the subsequent avionics size, weight, and power that it would take to operate such a system.

3.1.1 Nominal Operations

For dormancy operations, lack of crew intervention is a given, but some combination of ground and autonomous operations should still be available. For Deep Space Exploration class vehicles though, some of these prior strategies will no longer be employable. Ground intervention will still be available, assuming no permanent loss of communications, but communication latencies with the ground will increase. Still, there are expected to largely be two classes or types of intervention. One would be associated with a nominal dormancy state or subsystem states that transition to a safe state until such time that the ground can intervene to do the appropriate diagnosis to transition back to a nominal state. The nominal state could be reacquired via reconfiguration of subsystems or by reestablishing appropriate redundancy via such means as robotic remove and replace (R&R) (as well as potentially repair).

For mission phases in which the TTC is expected to allow ground intervention to bring a failure out of safe mode back to a nominal state, many of the avionics boxes that support the subsystems could be single string with a watchdog timer that allows transition to another redundant box upon hard failure of the primary box. This would be expected for the more quiescent phases such as during transit phases. The driver would be that in order to conserve resources, many redundant items will be powered off. Subsystems not needed at all would be expected to be powered off. For the more dynamic events, such as docking, a redundant box would expect to be powered on so that quick handover upon a failure could occur.

For nominal dormant operations, the following avionics assumptions in Table 12 should be made (Avionics in support of human interfaces, such as displays, audio, etc. are assumed to be off).

3.1.2 Transition Operations

3.1.2.1 Transitioning from Full Operations to Dormancy

Transitioning from full operations to dormancy is assumed to encompass largely crewed presence to uncrewed, dormant operations. Onboard Avionics can be broken into generally two areas; C&DH and Human Interfaces. C&DH encompasses the networks, processors, and instrumentation necessary to keep the vehicle operating. Human Interfaces are largely those avionics devices that support the crew, such as audio, video, displays and controls, and wearable technology.

Since the crew will not be present for dormancy operations, in order to save power and component life cycle considerations, it is expected that all the human interface related avionics equipment will be powered off as part of the crew departure. It is also assumed that planned crew departure would occur during ground communication opportunities, despite up to about 30 minutes in latency. Laptops and video cameras would be stowed. Fixed displays and audio, since they support Caution and Warning (C&W), would remain operational up to and through crew departure from the habitat. Wearable technology could be either stowed or transferred with the crew to the transit vehicle, depending upon intended usage. Once the crew has departed, ground commands would subsequently shutdown any remaining fixed displays and supporting audio.

C&DH, on the other hand, is critical to the Spacecraft survival in that it provides the means for other subsystems to perform their dormant operations. Some equipment is expected to be powered down to a minimal floor level in order to again, conserve power and conserve component life. Which actual devices to be powered down will depend upon the network and processing distributed configuration, which subsystems are supported by that configuration and their needs, and what their TTC needs are upon failure. The time-to-effect response will largely drive whether an immediate fail over to a redundant avionics component or a reboot of the existing component takes place. If it is a reboot of an existing component, a certain number of tries within the time-to-effect window would be allowed prior to switching over to a redundant component.

3.1.2.2 Transitioning from Dormancy to Full Operations

Transitioning from dormancy to full operation would begin just prior to crew arrival. Most of the C&DH avionics equipment that would be powered up from the dormant state in order to support the crew once they inhabited the habitat will be powered up in advance. This will largely be driven by the need to ensure that the various systems that are needed to support the crew are actually operational prior to the crew ingress. The avionics in support of rendezvous, proximity operations, and docking (RPOD) would need to occur first and prior to the beginning of those operations. Other systems activation, including fixed laptops and audio in support of C&W, would depend upon timeline availability and could occur prior to or after docking, but would need to occur prior to crew ingress. Laptops and wearable technology would be activated after ingress.

Table 12: Avionics and Software Assumptions for the DRM

Avionics Capability	**In orbit at Destination (Dormant period of 2 yrs)**
Planned Sensing, Monitoring, and Fault Detection Tasks	Support monitoring of subsystems (e.g., thermal, power, etc.) based upon their requirements during dormancy.
Planned Tasks During Maneuvers	Activation of subsystem components in support of propulsion (e.g., valve drivers, additional temp and pressure sensors, etc.) Deactivation when no longer needed for maneuver.
Planned Logistics Tasks	None
Planned Preventative Maintenance Tasks	None
Anticipated Corrective Maintenance Tasks (Repair)	ORU or LRU (i.e., card level) replacement. Would need to be robotics compatible.
Occupancy/Departure Preparation Tasks	Expand or decrease number of avionics related equipment based upon need of subsystem requirements in support of occupancy preparation or departure. Avionics human interfaces, such as audio and displays, are made available.

3.1.3 Contingency Operations

Historically for avionics, different architectures have been deployed in order to address different potential contingency scenarios. For vehicles that have had to deal with dynamic aspects of flight such as ascent and entry, a scheme that entailed a separate processor with different software was employed. For Shuttle, this was the Backup Flight Software that was resident in one of the General Purpose Computers. For Orion, this is the dynamic backup flight software that resides outside of the main computing system (Vehicle Management Computers [VMCs]) in a separate processing area (Vision Processing Units).

For the ISS, there is no backup flight software system, largely because the time-to-effect was significantly longer than the contingencies that could occur during dynamic flight. For dynamic flight, response times were often quicker than human response times and thus required an automated response. Historically for Shuttle, though, the primary avionics system software never needed to go to the backup flight software system in that the primary software worked as designed, largely due to extensive software development and testing. The backup flight software was largely covering for a software failure, not an avionics hardware failure (although simultaneous failure of all the general purpose computers operating primary software would have been a case that could have been covered, albeit a highly unlikely occurrence).

The working longer term ISS strategy was based upon primarily two assumptions, 1) that the ground had sufficient time and communications to do the necessary trouble shooting, and 2) that a failure would not preclude operations that would be necessary for ISS survival until the ground or crew could address the problem and get back into a more nominal state. This meant careful consideration as to what software functionality was placed in what avionics hardware, so as to ensure to the greatest degree possible of continued functionality until the problem could be resolved.

As an example, the pointing of the solar arrays toward the sun involves the motors that drive the solar array rotary joints to keep tracking the sun. Part of this function is based upon updates that came from the ISS command and control computers to maintain tracking the sun based upon orbital considerations such as location in orbit and vehicle attitude. If for some unlikely reason communication was lost with the command and control computers (unlikely because there are three of them), the ISS was designed so that the solar array rotary joints would keep turning so that near term, they could keep tracking the sun. Eventually, the uncertainties and biases associated with orbital and attitude dynamics would cause the pointing to drift from the actual sun location, but this is not immediate and is delayed (contrast to the case if the solar array motors decided not to move/track at all because of the loss of the pointing calculation information). This would buy additional time to keep from having to go immediately to the batteries, which over a certain amount of time would eventually become depleted, therefore allowing less time to resolve the problem. Had all the software functionality been placed in the command and control computers, this would not have been an option. Avionics is the focus here because software is meant to be adequately tested on the ground prior to being utilized onboard, so the software being the actual cause of the problem is limited. If it is, often options to go back to a previous working version of software exist, again assuming there is sufficient time-to-effect.

It is assumed that for Exploration vehicles a similar type of strategy will be employed. For vehicles that only reside in deep space transit or orbit, a backup flight software and processor will probably not be necessary. For vehicles that transit to and/or from a surface, this strategy may not be sufficient and a backup capacity such as a backup processor with backup software may be employed.

For nominal dormancy operations, there are potential scenarios that could play out, largely dependent upon the eventual architecture and time-to-effect. One would be associated with a nominal dormancy state or subsystem states that transition to a safe state until such time that the ground can intervene to do the appropriate diagnosis to transition back to a nominal state. The nominal state could be reacquired via reconfiguration of subsystems or by reestablishing appropriate redundancy via such means as robotic R&R (as well as potentially repair). The significant drivers here are the failure's TTC pitted against the ability for the ground to intervene before that time, including communication latency and communication outages as part of ground intervention allowable timeline. Also, depending upon the failure, transitioning back to a nominal dormancy state would need to occur before a critical transition dynamic event, such as autonomous rendezvous and docking (AR&D). For example, this means an event that brings one vehicle into a safe state that could theoretically be brought back to a nominal state in approximately an hour might not be the appropriate course of action for two vehicles that needed to be in a nominal dormant state to support AR&D that were to occur in a few minutes.

This would often be the case in what is considered transitional operations. Either the probability of such an event is considered so low that it is deemed an “acceptable risk,” or if not, it becomes a candidate for autonomous operation for the nominal state recovery. The other class of autonomous operations would be those in which ground intervention was never practical to begin with. An example would be a critical docking event in which some of the supporting docking sub-systems would need assurance to be in a nominal state as part of the final docking maneuver. For failures that would attempt to take the sub-system out of its nominal state, autonomous FDIR would be expected to take it back to a nominal state that would support the docking event. After the docking event is completed, eventual ground and/or robotic intervention could address reestablishing redundancy and/or repair. This assumes that the appropriate data has been collected in order to facilitate the correct course of action.

Avionics hardware that is considered operationally critical should be considered to have sufficient redundancy so as to enter a fail-operational state based upon a critical failure.   Payloads and science, which theoretically can be sacrificed in a worst case scenario would be expected to be able to enter a safing condition, so as to not impact other systems. Often this can be as simple as just being powered off. A fail-safe state is a potential option for some avionics, but implies significant characterization of the mission profile to assess when a fail operational vs. a fail-safe condition is warranted. Often time, the design for avionics will need to address the fail operational condition for critical functions anyway, negating much consideration toward the fail-safe scenarios since a fail operational state would already exist.

3.1.4 Preventative Maintenance and Logistics Support

For Avionics, typical strategies employed in the past have been flying the avionics component to failure, therefore maintenance is really more of an R&R strategy upon failure. Tools to perform this task are expected to be part of the intravehicular activity (IVA) tool set provided. From a sparing strategy, critical items would be expected to have a backup capacity, with the number being predicated upon analysis and associated expected reliability. Non-critical items would likewise be based upon analysis, with no spares being an option to the other end of the spectrum to a few, depending upon the desire to maintain the non-critical function in case of failure. Reliability, failure tolerance, and risk tolerance would determine the need for robotic IVA in case of failure during dormancy operations. IVA to date has been handled via human intervention, but extended unmanned operations would likely drive a reassessment of this approach with a potential outcome of internal avionics being designed to be robotically compatible. For external avionics, the same strategies would imply, with the expectation that all external avionics would be robotically compatible anyway to avoid the need for unplanned EVAs. A potential exception might be noncritical avionics components in support of lower value science, for example. Considering the associated cost of putting this science in a deep exploration mission, it is still more likely that external avionics would be robotically compatible for convenience than not because of the return on investment for making it so.

3.1.5 Autonomous Functions Needed

The time to criticality of the avionics system will be architecture dependent and will depend upon the robustness of the overall architecture to adjust and adapt. The ability to address this will be dependent upon backup capabilities as well as hardware and software distribution within the architecture.

As with most other systems, much of the work in monitoring and changing states and diagnosing and recovering from failures has been handled by the ground. As such, there are several autonomy gaps discussed in this section.

Faults include soft faults (unexpected crash/halt of programs, unexpected output of programs due to single event upset [SEU] or permanent fault) and hard faults (loss of a computer, network switch, mass storage device). Faults in other systems may have consequences on the avionics. Power and thermal faults are primary examples.

3.1.5.1 Robustness of Hardware

Because of resupply and latency concerns, higher reliability is going to be needed to reduce the need for FDIR to occur. This can be addressed through internal design redundancy and creative degraded mode operations.

The only realistic way to increase the time between when failures occur would be through increased design reliability. Avionics hardware faults will most likely be due to some environmental condition such as radiation. Most of these impacts can be addressed by using Error Correction Code (ECC) Memory (e.g. 4-bit ECC NAND Flash) to correct bit flips. Technologies like this will keep the software healthy and assist in supporting standard schemes to refresh executable images and data from persistent memory on a periodic basis. Solid State Memory capable of protecting itself allows for the preservation of software programs and data, guaranteeing the software can be recovered. Research on radiation tolerant chip designs such as redundant radiation tolerant designs that better handle single event effects is currently ongoing. The ability of the avionics hardware to correct itself fits well with the approach of a layered FDIR capability in the various software subsystems.

Developments are needed to minimize radiation susceptibility of the most vulnerable, high-risk components. This implies identifying the components whose loss would lead to the loss of the avionics hardware. Good examples of this are power supplies and network switches. This leads to the idea of selective radiation hardening of components versus radiation hardening of entire processing boards. This mix of radiation hardened and radiation tolerant hardware may be a way to save mass and power, and increase processing throughput.

Other ways to design the hardware to prevent hard failures or faults caused by environmental conditions exist. One is to leverage experience, techniques, and technology from the deep space explorers, ISS subsystems, etc. to address early hardware mortality and certify long-life components.

Watchdog overcurrent protection is another technique. Other methods to extend hardware component end-life should be researched.

3.1.5.2 Fail Operational/Fail Safe Design

Initial critical failures should largely be based upon the assumption of "fail operational," usually handled through redundancy. Subsequent failures in the same functional area should address a “fail safe” option, which could imply non-function for non-critical applications until a later time. For some critical applications, this could imply a non-operational status until functionality could be restored, assuming functionality is restored prior to the application’s TTC.

Replicating flight computers, network switches, or both, also allows fault-masking capabilities such as voting and lock-step redundancy, either software or hardware based; it is assumed such capabilities are built into the avionics and software, and these capabilities are foundational avionics design principles. However, voting may identify faults that can be recovered by autonomous technology such as FDIR.

Another take on fault response considers computing availability. The computing availability or “up time” challenge is a different take on the average time an avionics hardware asset is available. This concept aggregates all available computing assets and bases the availability on the entire set of avionics hardware that the flight software has to execute on. The implication is that any software application is capable of running on any processor as long as that processor is networked in relation to all other computing assets. The objective is 100% availability of the software system’s processing capability, rather than 100% hardware up time. Redundancy or cross-strapping of centralized processors versus a “pool” of processors distributed throughout the subsystems is a gap that could be traded.

A network topology is set up to provide this soft “fail-over” option. For instance, the system must be able to determine whether a processor board is no longer viable and therefore be able to transfer all the software operations to another board or to distribute them amongst other processors. The concept is to provide a set of networked processors that are transfer targets, but are not necessarily flagged as redundant for any software application. An approach such as this provides the autonomous functions with greater flexibility in determining how an avionics hardware fault or failure is to be addressed, while the failed hardware asset is being recovered or possibly replaced.

Subsystem compatible processors and platforms or “one size fits all” processor boards could be utilized. A common hardware set across all subsystems facilitates parts replacement since all computing resources are based on a common processing asset. Not only does this simplify robotic replacement and logistics, it also provides a path to upgrade all processing boards via logistics missions if the avionics hardware is nearing end-of-life.

3.1.5.3 Autonomous Fault Detection, Isolation, and Recovery

The avionics system state has typically been monitored from the ground. For short periods of no communications, failures could be addressed by autonomous FDIR. Upon acquisition of signal, the ground could work toward obtaining a better, more desired state. Since significantly longer loss of communication scenarios and time delays are expected, this implies that the "obtaining a better, more desired state” needs to be addressed onboard.

In the event of a soft or hard fault, assuming redundant avionics hardware (cold or hot spare computer, network switch, mass storage, etc.) is available, the first response is to switch to the redundant system. For dormant operations, one option would be to remain on the switched over redundant component/string until such time that the ground could reestablish communications with the dormant vehicle. This implies high component reliability and an acceptably low risk threshold that the last remaining string can work until redundancy can be reestablished by ground intervention. Critical avionics system performance parameters that would typically be available to the ground during continuous communication (ignoring latency) would be expected to be recorded on board for eventual downlink to the ground upon reacquisition of communications after the extended outage. It is expected that there would be some simple autonomy to recover from soft faults, such as an autonomous reboot of the component to see if it can be revived and a handback could occur. If the component cannot be reactivated into an acceptable state, the redundant component would remain in control until which time the ground could intervene and restore the failed component to an operational state. Another option is to accommodate all or a large portion of soft failures via subsystem automation during dormant, extended loss of communication scenarios.

In principle, hard failures could be addressed robotically via R&R operations. While this seems ideal to accommodate reestablishing redundancy as quickly as feasible via automation during extended loss of communications, the drawback is the complexity of this solution. If reliability were a concern, then it is questionable that reliability was adequate to begin with to accommodate the long duration of the mission on a single string. Depending upon the architecture, if the time to effect for a critical operation is of a duration that a “cold spare” can be brought up in the interim, then this part of the architecture could even assist in any reliability concerns by allowing the ground to alternate between prime and backup and consequently adjust the duty cycle between the components.

3.1.5.4 Smart Telemetry

Due to the long command and telemetry latency and outage periods, acquisition of communications will mean that the ground will need to understand any changes to the last known configuration, including any onboard autonomous FDIR actions required during the LOC time period.

Because of the potentially long delays in telemetry, the implications are that stored telemetry would subsequently be made available to the ground, so that whatever action or changes occurred could be assessed. Considering the potential for excessively large volumes of stored telemetry that would need to be sorted through if everything were collected, an intelligent system that adjusts the telemetry collection based upon the onboard states and changes would be ideal. Barring a failure that would cause or result in a change in the avionics state, avionics would not be expected to be a major user of this telemetry adjustment, at least in the context of its own systems being monitored, but wouldn’t be precluded either should the state change need to occur as a result of a failure or scheduled change in its planned operation.

Since during long duration loss of communications situations the ground cannot intervene, there could be a good argument for adjusting what is being recorded onboard based upon what a centralized intelligent system is seeing. Autonomous FDIR would cover desired system behavior in order to remain operational, but the additional insight provided by the adjustments could assist in future exploration vehicles' avionics design.

3.1.5.5 Script or Procedure Execution Monitoring and Logging

Because of significant time delays, critical transition activities will require some degree of autonomy, somewhat dependent upon the pace of desired transition in conjunction with the time latency. Failures in the state change procedure may require an autonomous recovery option, such as going back to the initial state until the issue can be remedied by the ground. For operations that must occur due to short time to criticalities, workarounds or autonomous functions to get to the new state would be necessary.

Options for dormancy operations include scripted commands to get to a new avionics state. For avionics components, these would be expected to be rare in the context of intently powering up and down components. There are potential exceptions, such as planning for the crew coming back to a dormant station from Mars surface in which various components that typically would be powered off may be powered up. Operationally though, it is possible that this would largely be delegated to some combination of remote crew and/or ground commanding to the dormant station with holds and checkpoints being assessed before moving on to the next phase. Barring an unplanned avionics state based upon a failure, this would probably be the more likely outcome. Planned changes in avionics state (again, excluding internal software change activity) would probably be the exception, not the norm.

Monitoring and logging script execution in support of state transitions will be necessary in order to determine the successful execution of steps associated with the transition as well as verification that the final transition has been successful.

3.1.5.6 Prognostics and Trending

Currently, prognostics is largely accomplished by ground controllers monitoring for trending behavior. Due to extended LOC periods, trending may need to occur onboard with adjustments to what is being monitored based upon changes to the expected system state. This could include requiring revised or alternative remedies based upon avionics trending information, which includes degraded mode operations.

Avionics has typically adopted an operational paradigm where components have been flown to failure. While there may be a need to monitor aspects onboard for later downlink and ground evaluation, there may not be much of a driving case for predictive or prescriptive behaviors being done onboard based upon long term monitoring. For avionics, fault detection attributes and aspects should already be defined to sufficient detail that the behavior of the component is either fully functional, degraded within predetermined bounds, or failed.

3.1.6 Development Plan

Autonomous system development for avionics has two focuses. First, the system design must consider several things with respect to increasing the time to criticality for any failure, namely robustness, failure strategies, and architecture for critical avionics fail-over or repair. Second, there are several autonomous algorithms that need to be advanced to handle fault detection, isolation and recovery, data collection, and even prognostics. While several developments in this area are ongoing, this section will focus on all priorities for advancement for space applications.

3.1.6.1 Systems Engineering Design for Avionics

3.1.6.1.1 Robust High Performance Avionics

In order to increase the reliability for long-duration spaceflight, avionics components must be designed for extended periods in high radiation environments. The reliability of individual components reduces the need for elaborate failure responses, since the likelihood of multiple failures is smaller. Work is in progress on radiation tolerant chip designs that can better handle SEUs. The High Performance Spaceflight Computing project is focused on drastically improving the processing capability for space missions, which is needed for both reliability and for the increased need for processing due to the sensor processing and autonomous functions that will be required. Further and accelerated investments in this area is essential for success of the design reference mission, as this hardware must be in place and available several years before the spacecraft is launched, in the early stages of design.

3.1.6.1.2 Failure Strategies and Repair

Architecture design for avionics is important for shaping the failure response of the spacecraft. In conjunction with (and highly dependent on) the increased component reliability, the strategy for dealing with failed components must be part of the overall design of the autonomous system. Autonomous FDIR is dependent on this strategy, as complexity of the algorithms are contingent on fail-operational, fail-safe, or other strategies. There is a mass and complexity trade, as well, on whether avionics components that experience hard failures can or need to be replaced or repaired. Investments into studies on failure responses of critical systems in a dormant spacecraft should be conducted to learn the requirements for reliability of components, spare management, robotic repair options, and risk postures early in the technology development process.

3.1.6.2 Autonomous Algorithm Development for Avionics

3.1.6.2.1 Fault Detection and Isolation

While this is generally described as part of system health management, which is covered in the Development Plan for Common Technologies section, the difficulty that presents itself for Avionics is the need for remote handling of errors. Generally, when an avionics component fails, the failure must be determined and corrected from a different avionics component. Currently, ground is responsible for deciding what failure has occurred and how to recover it. Future development is needed to determine the best distributed algorithms for determining when a fault occurs, where it occurs, and which hardware has the command authority to recover the fault. This ability is highly dependent on the avionics architecture, as described in the previous section, and should be developed concurrently.

3.1.6.2.2 Autonomous Procedure Execution

Recovery of errors will require several steps that must be accomplished with determination of the success or failure of each execution. This function will only be used when the time to criticality is such that the ground cannot effectively be in the loop for the recovery. As such, the procedure execution ability must be sufficiently intelligent to be able to reconfigure the system to safely increase the time to criticality until ground control can assess the situation.

3.1.6.2.3 Smart Telemetry Management for Situational Awareness

For all systems that require data sent to ground controllers in order to make them aware of the situation surrounding an anomaly or fault, some intelligence will be required to sort through the data in order to optimize the bandwidth available. It is likely impossible to encode what data to save and send in every situation, so it is important to have a system that can adapt to new circumstances in an appropriate manner. In addition to sending the appropriate data to Earth to allow ground controllers insight into a situation, it is also important to frame the data in a way that can be easily parsed and analyzed by the human controllers. Research is required in learning systems, cognitive systems, and explainable models for adaptive controllers in order to accomplish this important function.

3.1.7 Criticality and Redundancy

The Avionics systems are necessary to ensure the functioning of the vehicle in the absence of crew members on board as well as when they are present. When the crew is present, additional avionics related items are expected to be active in support of the crew such as audible alarms for C&W as well as displays and controls. The criticality of the avionics systems depends on the criticality of the task they are commanded to complete. Typically, the architectural development of the avionics systems reflects this reality. Often, because many software tasks can be aggregated in one avionics box, failure of said box is often of a short time criticality driven by the shortest time to criticality of the software supporting task being performed. Because of this, as a general rule, those avionics components that engage in time critical catastrophic related activities would expect to be 1R3 (as defined in Table 13). This does not imply, however, that catastrophic activities are the only activities occurring in the box, so some operations could have excessive redundancy capability when compared to the actual need out of convenience associated with aggregating tasks in the least amount of hardware needed. Because of the cost and national asset status of the ISS, a decision was made that even though the crew could escape the ISS, loss of the vehicle was considered catastrophic and drove a two failure tolerant design in some instances. This would be a difference when compared to the single failure tolerant vehicle design of many interplanetary probes. Mission Critical tasks would be expected to be at least 2R2 (defined in Table 13) in order to ensure or optimize mission success. Where loss or failure is considered tolerable (e.g., some science experiments), a Crit 3 designation could be invoked where there could potentially be no redundancy and failure would result in loss of functionality. This may be an acceptable posture until which time that crew is present and could restore functionality through replacement and repair.

There is often a trade-off between robustness and complexity in system design. Some strategies attempt to reduce weight, for example, by providing internal redundancy. An example would be Orion's VMCs. The design has redundant flight control modules within each VMC so that a complete failure in one would result in the 2 remaining flight modules for this particular catastrophic function still being available in the other VMC. Other modules in the VMC were singular, meaning that there were only two identical modules total between the two VMCs. Using the previous VMC failure, only one module would be remaining in the remaining functioning VMC after failure. As such, careful consideration and design of control and safety systems are necessary to ensure avionics with the proper robustness, fault tolerance, and redundancy for the tasks required of them. Table 13 outlines brief criticality definitions as well as how avionics would typically attempt to address them.

Table 13: Criticality Definitions for Avionics

Designation	Definition	Avionics Response
1	Single failure that could result in loss of life, permanently disabling injury, or loss of vehicle.	From an Avionics perspective, there should not be any single failure that could result in loss of life because it can be designed out through redundancy or augmented with software through fault containment regions for undesired behavior.
1R#	Redundant hardware that, if all failed, could cause loss of life, permanently disabling injury, or loss of vehicle. A number (#) is used to indicate the number of failures required for complete system failure (1R2, one failure tolerant system; 1R3, two failure tolerant system, etc.).	All critical avionics would expect to be at a minimum of 1R2 and preferably 1R3. The difference between acceptability of one over the other would entail attributes such as overall reliability or time to effect for robotics. If a 1R2 type failure occurs, then consideration to bring back to nominal 1R2 levels are weighed against the mass, power, and thermal penalties of three strings versus two as well as the associated risk tolerance for being on one string until which time the redundant string could be brought back up will need to be considered. Being 1R2 is not without precedent from design perspective for potential loss of vehicle in that the ISS program recently replaced, under contingency EVA situations, one of two External MDMs that communicate with critical components out on the ISS truss.
1S	Failure of emergency equipment and systems, hazard monitoring systems, or other hardware (e.g., burst disks, relief valves, communication devices) that could result in loss of or degraded capability to detect, combat, isolate, or operate when needed during a hazardous condition	Not a real avionics designation. Addressed primarily by 1R# for Avionics.

	to preserve the life of the crew, prevent permanently disabling injury, or prevent the loss of the vehicle.	
1SR	Redundant 1S hardware that, if all failed, could result in loss of or degraded capability to detect, combat, isolate or operate when needed during a hazardous condition to preserve the life of the crew, prevent permanently disabling injury, or prevent the loss of the vehicle.	Not a real avionics designation. Addressed by 1R#.
2	Single failure that could result in a loss of mission or personnel injury or occupational illness requiring definitive/specialty hospital/medical treatment that results in loss of mission.	A single failure should not result in loss of mission or personnel injury or occupational illness. Would expect to be at least 2R2.
2R	Redundant hardware item that, if all failed, could cause a loss of mission or personnel injury or occupational illness requiring definitive/specialty hospital/medical treatment that results in loss of mission.	Expectation is that the design will be at least single failure tolerant against loss of mission.
3	All other failures.	Expect single instances of Avionics components to often be the norm where critical functionality or mission success is not at stake.

3.2 Communications

Communications is the subsystem that transfers all information to and from the vehicle. It conveys all state information from other subsystems to Earth and surface operations, and provides a platform to command these other subsystems. It also functions as a data source for other systems, e.g., Navigation derives parameters from the communication system. It is the bridge between Earth operations and the crew engaged in space and/or surface operations.

3.2.1 Nominal Operations

The following use cases during nominal dormancy operations are needed:

- Commands from Earth-to-vehicle
- Telemetry from vehicle-to-Earth
- Logistics communications, e.g., software updates, file uploads, etc. from Earth

- Health and status telemetry for vehicle subsystems from vehicle-to-Earth
- Engineering and science experiment data from vehicle-to-Earth
- Independent, high-reliability communication for emergency and safing situations
- Earth-to-vehicle-to-surface-assets commands
- Surface-assets-to-vehicle-to-Earth telemetry (possibly data rate adjusted to accommodate different data sources)
- Full duplex vehicle-to-vehicle communications

3.2.2 Transition Operations

Dormant operations are expected to be less demanding on the communication system since onboard systems are expected to be producing and consuming less data in dormant operations than in crewed operations. When there is crew onboard, the vehicle-to-Earth communications is expected 24/7. However, when the vehicle is in dormant operations, communications with Earth can be scheduled for a few hours a day or week, depending on the data transfer needs. As such, it is expected that the power requirements of the communication system could be at a reduced level in dormant operations, however, all links can be activated at any time. One important factor in determining the power profile and dormant communication system requirements is that the relay link to surface assets is likely to be more active than while in nominal operations since crew is expected to be active on the surface, and may be producing and consuming more data than while the vehicle has crew onboard.

3.2.2.1 Transition to Dormant Operations

Transitioning the communication system to dormant operations will consist of facilitating the transition of other subsystems as they go into dormant operations, supporting the commanding, telemetry, and other logistical communication requirements of these subsystems. This transition period could drive communication system requirements, depending upon required data rates, and the timeframe in which the transition into a dormant state needs to happen. As a baseline, data rates and times for transitioning systems for past robotic Mars missions will be leveraged and scaled to the number of links and data types required for human-rated vehicles. During the transition, all links for commanding, telemetry, relay, and emergency communications will be utilized.

Following the transition of other subsystems into a dormant state, the demands on the communication system will be reduced. The communication system can be put into a dormant state, which is still active, i.e., in receive mode, but requires less power consumption. Subsystems making use of the communication system will again drive requirements, but since the data rate requirements should be at a reduced level, except on the relay link, some power savings should be realizable. Transitioning to this dormant state will require transitioning over to this dormant state. All links will remain in receive mode in their dormant state, lower power state, and periodically turn on their transmitter to report health and status.

3.2.2.2 Transition Out of Dormant Operations

Moving from dormant to nominal operations consists of first switching out of the dormant state, following a similar make-before-break strategy, and into a nominal, higher-power, communication system state. As each link is brought into a nominal operations state, a complete checkout of the underlying subsystems would occur, testing subsystem health and status metrics to ensure that it operates within expected bounds.

3.2.3 Contingency Operations

Space vehicle core communication systems nominal operations vary from spacecraft to spacecraft. For example, International Space Station (ISS) core communication system is continuously transmitting/receiving data. If there is no data to send, fill data is used to fill up the link. Space Shuttle core communication system also operated in a similar way. In these situations, if the onboard communication system does not receive a signal for an extended period of time (due to a failure, mis-match in configuration, etc.), the communication system falls back to a pre-defined, known default state. The ground controllers, as part of the contingency operations, can try to re-establish communications by using this known default configuration.

Other spacecraft have scheduled communications with the ground (their mission control). This could be once a day, once a week, at a pre-specified time, etc. There are different contingency scenarios to go with the scheduled communication operations. A way to handle contingency and/or impromptu communication operations is for the onboard core communication system to be in a known "receive" mode until it receives a command/data from the ground. This command can start nominal communications or contingency communication recovery based on the situation.

3.2.4 Preventative Maintenance and Logistics Support

No preventative maintenance or logistics support is expected during dormant spacecraft operations.

3.2.5 Autonomous Functions Needed

Three major autonomy gaps have been identified for the communication subsystem, autonomous re-configuration of onboard communications and ground stations, direct space-to-space communications with radiometric tracking, and networked communications for vehicle to EVA/free flyer/sensor.

3.2.5.1 Autonomous Re-configuration of Vehicle and Ground Stations Communication System Parameters

Autonomous re-configuration of ground stations is needed to sense changes in onboard communications system configurations (power, data rate, modulation, frequency, etc.). Currently these functions are completed manually. The ground control center identifies the loss of communications and sends a configuration change request to the ground station based on what the pre-defined FDIR protocols are. This capability is necessary to support autonomy functions not only onboard the vehicle but also provide additional autonomy in ground control operations. By being able to autonomously change configuration of the onboard communication system, it enables the space vehicle to recover faster from loss of communications, as well as provides the capability to adapt to changing channel conditions. For example, if the channel is strong, then the communications system can adaptively increase its data rate or lower its transmit power. Conversely, if the channel is degraded, the communications system can back down the data rates, change to more efficient modulation schemes, increase its transmit power, etc.

Communications between the space vehicle and the ground control center provides the critical life-line to transfer information and send updates and commands. When there is a loss of communications onboard the space vehicle, as part of its FDIR, it drops to a default configuration. Likewise, when there is a configuration change in the communications system, the ground station does not automatically detect nor change its state to the new configuration. The mission ops/control center operators have to detect the loss of communication or determine the configuration change and have to request the ground station to change the settings to the new configuration. If the ground stations are upgraded to automatically and adaptively sense the changes in the received signal and reconfigure its settings, this would minimize the duration of the loss of communications as well as provide some autonomy in the ground station and control center operations.

This capability would require both system design and subsystem automation to enable this function. This has not been done before now largely due to the cost of designing and upgrading the ground stations. Also, trust is an issue; controllers want to be in the loop as part of configuring the communications system. Because the onboard communication system can be in numerous configurations, it can get very complex to autonomously go through each combination or permutation to determine what the current configuration might be. The solution to this problem must be capable of handling the various permutations in a provably correct manner.

3.2.5.2 Direct Space-to-Space Communications with Radiometric Tracking

Direct space-to-space communications with radiometric tracking between space vehicles irrespective of the communications link with Earth would provide many benefits for this design reference mission. A communication link which also provides radiometric tracking (providing range and range rate) is needed to support ability for the space vehicles to autonomously rendezvous and dock without requiring a communications link with Earth. This is especially true as missions are conducted beyond where global positioning system (GPS)-type navigation is currently available.

The onboard communication systems need the ability to operate as a “ground station” to generate the ranging signal and use the coherently transmitted received signal to calculate the range and range rate. In addition, the radio frequency (RF) systems need to be able to operate in reverse band in addition to nominal transmit/receive frequency band.

GPS currently provides this capability for ISS-related missions. This has not been needed before since rendezvous and docking between two vehicles outside of low latency Earth communications is rare to non-existent. Many challenges to providing this autonomous functionality exist, particularly in system design. For example, reverse banding requires complex RF and antenna systems.

3.2.5.3 Networked Communications

EVAs, free flying cameras and other sensors will be operating around the space vehicle during nominal and contingency operations. A communication system will be needed to support these functions. This communication system needs to be standards based for interoperability. It should also support smart networked communications where the system can autonomously sense which users it can communicate with, determine how to route data, form the network, and reconfigure in the dynamically changing environment. This requires further development and standardization activities.

3.2.6 Development Plan

The three autonomy needs identified for the communications subsystem involve mostly systems engineering and design development work to achieve, but some algorithmic advances will be necessary as well. This section is divided by these two topics.

3.2.6.1 Systems Engineering for Communications Autonomy

The three autonomy needs, and in particular the networked communications to support vehicle to EVA/free flyer/etc. communication system, require investments into developing the systems in question. The space-to-space communication system is currently implemented in Orion, but it has not been tested or proven. It needs to flown in conjunction with another vehicle, and performance validated during RPOD. For the autonomy in the ground station, further sensing abilities are required to get the information needed to make the adjustments automatically. Developments in creating the necessary sensing capabilities are required investments.

3.2.6.2 Algorithm Advancements for Communications Autonomy

The algorithms needed to add autonomy to the spacecraft-ground station system include online model adaptation, planning, and verification and validation (V&V) of autonomous systems. These topics are discussed below.

3.2.6.2.1 Online Model Adaptation

As the spacecraft moves further into unknown environments and conditions, differences in modeled performance and failures are expected in the communications system. As such, an autonomous ground station capable of reconfiguring when spacecraft communications degrade or fail will cause a change to the model for the communications. This is different than planning, in that there are many inter-related knobs to turn, and the adaptation of the model is essential to the determination of the plan for which adjustments should be made. The set of technology solutions that could make online adaptations of models includes the various learning and cognitive reasoning techniques that are under research currently. Commercial investment in these technologies are strong; NASA's investment should focus on the technologies that not only allow for online adaptations, but also those that can accommodate guarantees on performance and safety constraints.

3.2.6.2.2 Planning

As adaptions need to be made by the ground stations, planning is required to create a set of adjustments that should be made. These adjustments will be informed by the model and by constraints present in the system, timeline, or other needs. Constraint-based task planning will be required to generate the activities and adjustments that will ensure a successful acquisition of sufficient communication signal. Further treatment of the developments needed for planning can be found in the Development Plan for Common Technologies section, Section 3.13.3.

3.2.6.2.3 Verification and Validation of Autonomous Systems
The most important function of any autonomy associated with the communications systems is the ability to ensure that communication will always be able to be established. The loss of communications with the spacecraft is extremely critical, and trust must be gained by any autonomous functionality prior to and during deployment. A discussion on the technical challenges in this can be found in Section 3.13.4.

3.2.7 Criticality and Redundancy
The communication system provides the critical life-line between the space vehicle and control center on Earth by enabling the transfer of data and commands between the two. Data can be health and status telemetry, engineering data, audio, C&W data, imagery, crew communications (email, internet access, etc.), file transfers, software uploads, etc. The operation of the communication system is critical to mission success and safety of the crew. Even as missions are further away and communication latencies are too high to enable real time audio/commanding/etc., getting the information back to Earth is still necessary to complete the mission objectives and ensure the safety of the crew.

Redundancy of the communication systems is provided in different ways: having two redundant strings; having two or more different systems and duplicating "critical" data on both; having different communication paths, etc. The Space Shuttle and International Space Station utilize(d) all of the above redundancies for their communication systems. For example, the ISS has two strings for the S-band communication system, has a Ku-band system that duplicates the core data from the S-band system, can relay some of its data using the Russian communications system, and has a voice only UHF/VHF system.

As more autonomy is designed into the space vehicles, ensuring that the communication system is robust is critical not for not only sending up commands, software loads and getting telemetry back, but also to learn how the vehicle responds to the onboard autonomy and to learn what needs to be improved upon.

3.3 Environmental Control and Life Support System
The ability to support human beings and keep them healthy and highly productive is critical to the success of any human exploration endeavor. Spacecraft life support systems, depicted in Figure 4, are responsible for maintaining a healthy environment for the crew by removing contaminants, providing resources, and controlling environmental parameters. One thing that differentiates the design of life support systems from many other spacecraft subsystems is that humans essentially need the same consumables and the same environment no matter where the spacecraft is going. Nevertheless, the outside environment, mission design, and vehicle constraints can result in the use of significantly different technologies or designs to accomplish a similar end goal for the crew.

Though the ECLSS is primarily designed to maintain a habitable, safe environment for the crew, it also provides important functions to support other vehicle systems during both crewed and dormant phases, ensures a safe, habitable environment upon the crew's return, and continues to operate through the next inhabited mission phase.

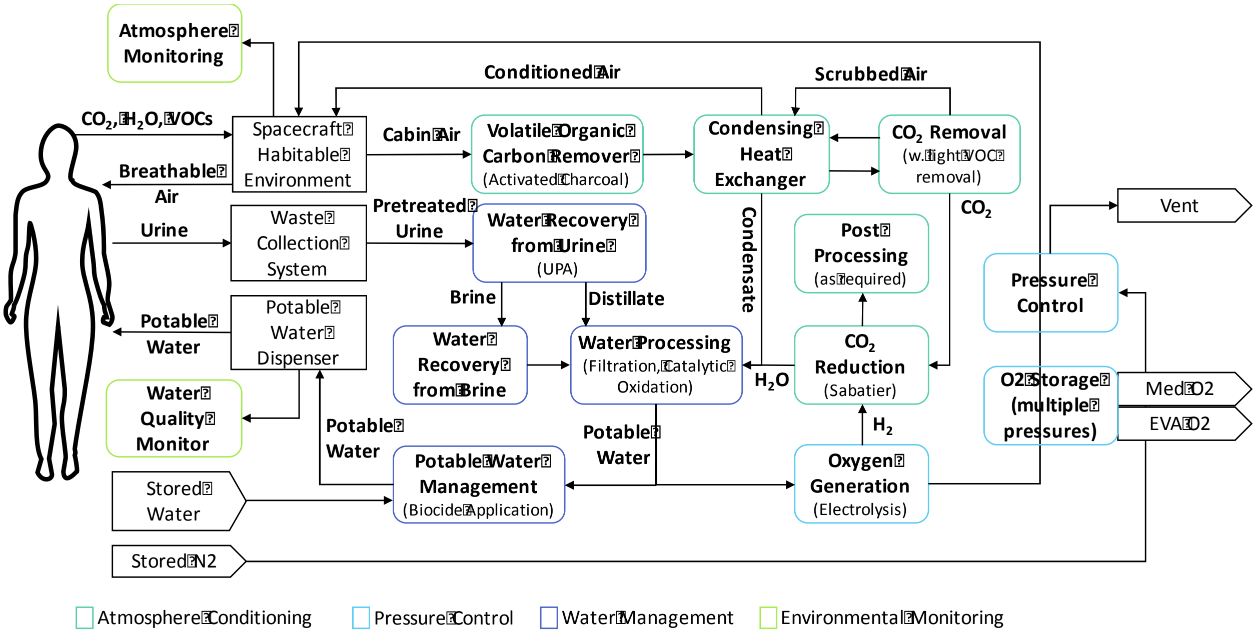
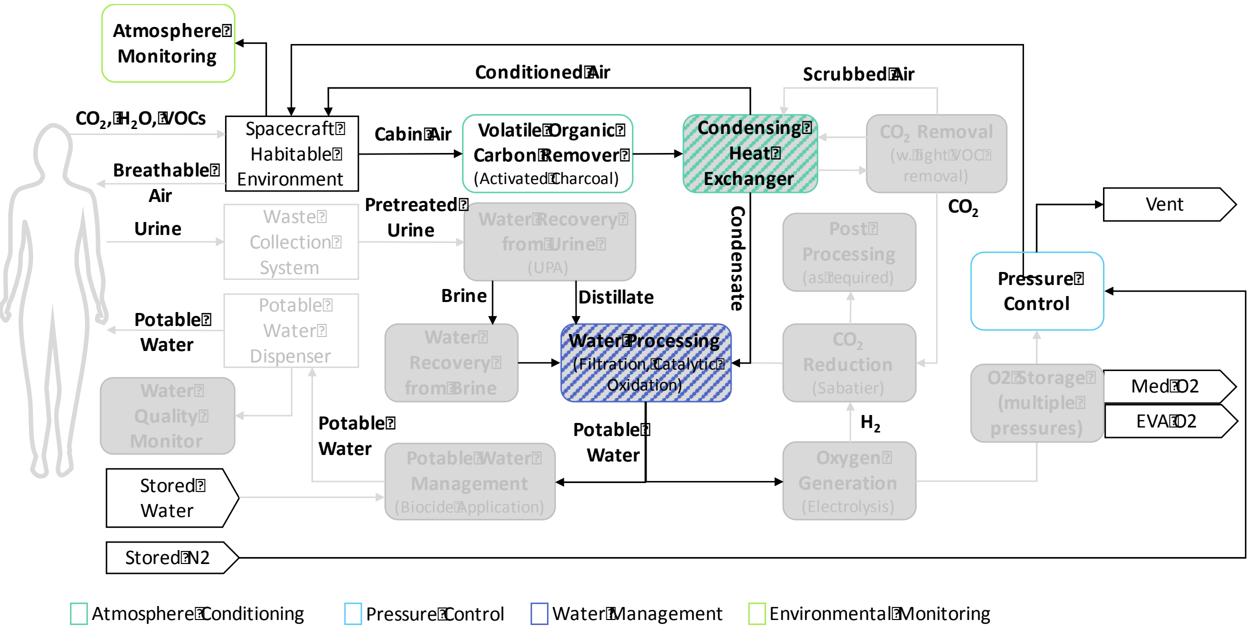

Atmosphere Monitoring
CO$_2$, H$_2$O, VOCs
Spacecraft Habitable Environment
Breathable Air
Urine
Waste Collection System
Potable Water
Potable Water Dispenser
Water Quality Monitor
Stored Water
Stored N2
Conditioned Air
Cabin Air
Volatile Organic Carbon Remover (Activated Charcoal)
Condensing Heat Exchanger
Scrubbed Air
CO$_2$ Removal (w. light VOC removal)
CO$_2$
Pretreated Urine
Water Recovery from Urine (UPA)
Brine
Distillate
Condensate
Water Recovery from Brine
Water Processing (Filtration, Catalytic Oxidation)
H$_2$O
Post Processing (as required)
CO$_2$ Reduction (Sabatier)
H$_2$
Potable Water
Potable Water
Potable Water Management (Biocide Application)
Oxygen Generation (Electrolysis)
Vent
Pressure Control
O2 Storage (multiple pressures)
Med O2
EVA O2
Atmosphere Conditioning
Pressure Control
Water Management
Environmental Monitoring

Figure 4: Simplified ECLSS Block Diagram for Deep Space Exploration Missions

3.3.1 Nominal Operations

The ECLSS provides minimal capability during uninhabited mission phases, shown in Figure 5. The required functions are limited to maintaining an environment in the spacecraft's pressurized volume that is sufficient to support the vehicle systems. Environmental parameters to be considered for dormant period maintenance include temperature, pressure, humidity, and O_2 partial pressure levels. It is expected that the control bands will be much wider than for crewed phases. Insight into system health and consumables quantities during dormancy will be key to ensure the safe return of the crew from Martian orbit.

The spacecraft modules should be isolated in order to minimize propagation of catastrophic events such as a fire or a non-recoverable pressure leak. Inter-module ventilation (IMV) must be provided to modules lacking intramodule ventilation and independent environmental monitoring capabilities. Atmospheric conditioning will remain active during the dormant phase, providing ventilation necessary for adequate air mixing to support environmental monitoring and thermal distribution. The temperature of the habitat will be allowed to drop to mitigate microbial growth; relative humidity will be greatly reduced to mitigate free water accumulation due to condensation at the lower temperatures. The Pressure Control System (PCS) will be in a standby mode in which the oxygen delivery valve is disabled and a minimum, dormant, total pressure is controlled with nitrogen only. Over-pressure relief is always active. The Water Management system will be inactive during dormancy except for automatic routines required to maintain potable water quality. Environmental Monitoring will remain active during dormancy.



Atmosphere Monitoring
Conditioned Air
Scrubbed Air
CO_2, H_2O, VOCs
Spacecraft Habitable Environment
Cabin Air
Volatile Organic Carbon Remover (Activated Charcoal)
Condensing Heat Exchanger
CO_2 Removal (w. light VOC removal)
Breathable Air
Urine
Waste Collection System
Pretreated Urine
Water Recovery from Urine (UPA)
CO_2
Vent
Condensate
Post Processing (as required)
Pressure Control
Brine
Distillate
Potable Water
Potable Water Dispenser
Water Recovery from Brine
Water Processing (Filtration, Catalytic Oxidation)
CO_2 Reduction (Sabatier)
O2 Storage (multiple pressures)
Med O2
EVA O2
Water Quality Monitor
Potable Water
Potable Water
H_2
Potable Water Management (Biocide Application)
Oxygen Generation (Electrolysis)
Stored Water
Stored N2
Atmosphere Conditioning
Pressure Control
Water Management
Environmental Monitoring

Figure 5: Dormant Configuration of the Exploration ECLSS.

3.3.2 Transition Operations

3.3.2.1 Preparing for Dormancy

The transition to dormant operations for the ECLSS will center on allowing the elements to remain in a standby state until the transition out of dormancy is initiated. Some functions will need to be maintained through the transition period in order to support the crew until they fully transition to another vehicle. If possible, modules should be isolated with IMV activated in order to minimize propagation of catastrophic events such as a fire or a non-recoverable pressure leak.

3.3.2.1.1 Atmosphere Conditioning

Nominal atmospheric conditions will be maintained until the crew is fully transitioned to the other vehicle. These control systems will be placed in a standby mode once the crew has departed the habitat. The transition of humidity control will be unique; humidity removal control will be required for some time after the crew has departed in order to mitigate condensation on habitat surfaces during temperature fluctuations.

3.3.2.1.2 Pressure Management

Pressure management will be placed in a standby mode that disables the oxygen delivery system and establishes a dormant mode pressure control band. Nominal leakage from the structure will cause a slow bleed down of pressure. A lower pressure limit may be driven by vehicle systems.

3.3.2.1.3 Water Management
Draw down of nominal water stores will occur ahead of a dormant period to reduce the amount of water remaining in the system. Where possible, water stores should be segregated from water recovery systems and the contaminated portions of the dispensing system. Water recovery systems will be flushed and treated with biocide prior to dormancy to minimize the biological contaminants in the system and then placed in a standby mode for the duration of the dormant period.

3.3.2.1.4 Environmental Monitoring
Environmental monitoring will remain active during the dormant phase. As mentioned previously, sufficient ventilation will be provided to support adequate mixing and sampling during dormancy.

3.3.2.2 Transition out of Dormancy
The transition out of dormancy will involve re-establishing a nominal habitable environment. Active atmospheric conditioning and pressure management will be re-established. A flush of the water management and water recovery systems will be performed to recycle the stagnant water. Any consumable elements that require refreshing will be installed by the crew.

Prior to crew return, a habitable environment must be established. This involves repressurizing the habitat, re-establishing a sufficient oxygen partial pressure, engaging the trace contaminant control system to remove off-gassed products that accumulated during the dormant period, and using the heaters to raise the temperature. A humidifier, which is not currently in the architecture, may be necessary to raise the humidity to required levels.

Prior to nominal operations, the remaining ECLSS assemblies will need to be activated and verified functional. The water recovery and management system will need to be returned to its operational configuration. The potable water system may need to be recycled to ensure adequate and consistent water quality.

3.3.3 Contingency Operations
Prior to crew arrival, a habitable environment must be achieved. If the habitable environment cannot be recovered, or verified, prior to transfer from the Mars Ascent Vehicle, then the crew must rely on some sort of protective equipment, up to and including suited portable life support capability for vacuum environments, to enter the habitat and manually recover capabilities.

This summary applies to the ECLSS induced contingency events. Nominal functionality of the spacecraft life support system is contingent on other system operations:

- Pressure Vessels: Excessive module leak rates, loss of critical resources from pressurized tanks
- Active thermal control: Loss of active thermal control
- Power: Loss of power
- Avionics: Loss of telemetry, loss of remote command capability, faulty commands to the ECLSS are not considered in this summary

3.3.3.1 Atmosphere Conditioning

The air revitalization system will be in a stand-by mode during dormancy. Failures could occur during pre-habitation ramp-up operations. Controller faults may be cleared remotely. Hardware faults would have to be resolved by the crew wearing the appropriate personal protection equipment (PPE) (e.g. suit, emergency mask, etc.) or by robotic assets. A failure of either intra-modular or inter-modular ventilation, which should be active during dormancy, would lead to poor air mixing and invalidate telemetry received from the Environmental Monitoring. In this case, either mobile robotic sensor packages could be deployed to understand the environment or the crew would have to assume a non-habitable environment. Robotic assets or crew wearing PPE will perform troubleshooting activities using portable monitors and recovery activities.

3.3.3.2 Pressure Management

A failure of the Pressure Management System during remote ramp-up could lead to a non-habitable environment. Depending on the failure and actual habitat pressure, either robotic maintenance is necessary or the crew may be required to don pressurized suits for troubleshooting and recovery activities. Some systems, such as the O_2 Generator, have inherent hazards that could lead to loss of the habitat. Hazard controls will have to be implemented into the design of these systems. These systems should not be operating during dormant periods. During ramp-up, these systems should not be initiated unless telemetry and remote command capability is established.

3.3.3.3 Water Management

The Water Management system would be as dry as possible during dormancy. A leak of water may be detected by continuous monitoring of the tank levels and cabin humidity. Biological growth may be detected with in-line monitoring and/or robotic sample analysis. Troubleshooting, repair, and recovery may be performed robotically or by the crew upon return.

3.3.3.4 Environmental Monitoring

In the case of a failure of the environmental monitoring system, either robotic maintenance and/or mobile sensors would be required, or the crew would have to assume a non-habitable environment and don emergency equipment for troubleshooting and recovery activities.

3.3.4 Preventative Maintenance and Logistics Support

The spacecraft ECLSS will require regular maintenance and spares replenishment during crewed operations. However, the ECLSS provides minimal capability during uninhabited mission phases and will, therefore, require little to no preventative maintenance during this time. Periodic monitoring of system status and parameters will provide insight into any contingency events and mission planning requirements (e.g. transition of certain spares or consumables from surface assets). Components that are known to fail to start after long periods of inactivity such as pumps and fans, may be periodically cycled. As part of preparation for dormancy, filters will be cleaned or replaced; without a significant source of particulates, the filters should support the full dormant period without planned maintenance. A trade study will need to be completed to choose between active microbial control during dormancy and flushing and reprocessing the Water Management System as part of the transition to crewed operations tasks.

3.3.5 Autonomous Functions Needed

The goal for exploration ECLSSs is to be as autonomous as possible to minimize crew time spent on repair or maintenance. This includes autonomous pressure management, controlling atmospheric temperature, humidity and oxygen levels, removing carbon dioxides and trace contaminants, and processing water. Periodic surface cleanliness and potable water quality currently involves crew intervention; automating these functions is desirable. It is also desirable to automate the transition processes and manage the dormant environment, though this may be performed via crew commands from the departing or approaching vehicle, or from mission operators on Earth. The following are the functions performed when the crew is not present.

3.3.5.1 Atmosphere Conditioning

The humidity control system will remain active after crew departure to dry the habitat air, mitigating condensation during the colder dormant environment. A humidifier may be needed prior to crew return as a habitable environment is re-established. Automated control of environmental heaters will maintain the dormant temperature and bring the environmental temperature back to habitable levels prior to crew return. Fans will remain operational during dormancy for thermal distribution, and for sufficient air mixing for effective air constituent's assessment. Trace Contaminant Control is not necessary during dormancy but will be re-activated to clear any off-gassed contaminants prior to crew return.

3.3.5.2 Pressure Management

The ECLSS will automatically control the habitat pressure to a lower dormant level. Oxygen will not be maintained during dormancy. Both nominal pressure and habitable oxygen level will be autonomously re-established prior to crew return.

3.3.5.3 Water Management

Potable water biocide dosing systems will remain operational during dormancy to mitigate microbial control. Periodic cycling of certain components may be performed to maintain functionality, such as cycling of rotational equipment and valves. Reprocessing of water may be performed prior to crew return. The water recovery system will be inactive during dormancy. Periodic cycling of certain components may be performed to maintain functionality, such as cycling of rotational equipment and valves. Reprocessing of water may be performed prior to crew return.

3.3.5.4 Environmental Monitoring

Contingency environmental monitoring will remain operational throughout dormancy. These monitoring algorithms will include leak detection in fluid systems such as valves / pipes (applicable to both gaseous/air loops and liquid loops), clog detection in fluid systems (biofilm, scum, solids buildup, etc.), detection of mechanical faults in actuated systems such as automatic valves, fans, pumps, detection of embedded computer faults (RIOs, little device controllers, as opposed to large avionics computers), detection of other types of hardware faults (e.g. the magnetron in the microwave for the Plasma Pyrolysis Assembly), and detection of sensor malfunctions, possibly with recalibration as a recovery action.

3.3.5.5 Other Needs

Algorithms to aid in the management of the spacecraft without crew present are required for long dormant periods. An example would be the prediction of when and where condensation might occur with localized heating/ventilation to mitigate. This could save heater power and fan life by running to worst case scenarios. Another need is the automatic detection and response to pressure leaks (where it is, and isolation or even repair of the offending structure), but this will be covered more completely in the Section 3.9. Robotic mobile sensing platforms to serve as a backup for failed portions of the Environmental Monitoring systems would be very useful. Also, there is a need for increased robustness of the ECLSS components as well as a method for repair of the system when hardware components fail. Finally, various condition-based maintenance algorithms may be required to detect filter, cat-bed, and other disposables change-out.

3.3.6 Development Plan

Dormancy is the most dramatically different mode for ECLSS hardware compared to its other normal operational modes with crew present. The ECLSS provides minimal functionality during dormancy to maintain a threshold environment (pressure, temperature, relative humidity, ventilation and environmental monitoring) to protect the remaining vehicle systems from extreme environmental conditions. These systems must be designed for full autonomy. Though constant operation of the non-essential ECLSS subsystems during dormancy is not planned, maintenance checks and transition operations are expected to drive the autonomy needs. As such, the system must be designed for fully autonomous operation for limited times. Criticality and time to criticality is reduced for the ECLSS during uncrewed phases. Much of the maintenance and even much of the transition period can occur during communication periods with ground control or with nearby crew, which somewhat reduces the need for algorithm advances for this subsystem. Developments needed for these autonomy gaps are split into two groups, the first being systems engineering considerations for the design of the ECLSS subsystems and the second being algorithmic and robotic developments needed for both the transition and contingency operations.

3.3.6.1 Systems Engineering for ECLSS Autonomy

The extensive requirements of the ECLSS drive the complexity of the system, with pumps, valves, pipes, tanks, filters, and sensors being extensively used to create a livable environment and provide potable water. As such, a full sensor network will be required to provide ground controllers and autonomous algorithms appropriate situational awareness to assess and react to events. A treatment of sensor network developments can be found in Section 3.13.1. Specifically for the ECLSS subsystem, the many different types of information that is needed to monitor the environment requires many different types of sensors. Research into consolidating the types of sensors needed or even into designing the overall system to reduce the number of sensor types needed (i.e., removing the potential of toxic gas release) is needed. Furthermore, trade studies that weigh the additional complexity of having mobile sensors versus an in-place sensor array would be beneficial for this system. Some of the sampling is currently done by the crew. System design must include methods for data collection during uninhabited periods, by using advanced sensors or data collection methods, or by using robotic sampling technologies.

A trade for the self-actuation of the many valves and quick disconnects and inline redundancy of the pumps and valves versus robotic actuation or repair options should be conducted. Similar studies are being conducted for related industries, in particular, the oil and gas industry is looking at robotic maintenance of off-shore platforms. Initial data suggests that robots are more cost effective than actuating each valve in these cases. Lessons learned from these studies should inform the studies NASA conducts for spacecraft maintenance and redundancy. If trades agree with the preliminary results that the oil industry is finding, investment into robotic manipulation and mobility is prudent to ensure the technology is robust for this critical application.

Other maintenance needed for ECLSS, such as surface cleaning and potable water quality checks, are currently managed by the crew. System design is required to either remove the need for outside manipulation required for these tasks or to design manipulation systems that can accomplish these tasks. Some developments for surface cleaning could involve dust repelling materials for surfaces, and potable water sensing could similarly be advanced through technology development. In lieu of these developments, robotic manipulators could be designed to test and clean surfaces as needed, and could provide mobile sensing for the potable water quality. A trade between the two technology paths should be conducted once the technology readiness level (TRL) of each are sufficiently advanced. Investment is needed to identify the appropriate solution to these problems.

3.3.6.2 Algorithmic Advances for ECLSS Autonomy

The algorithmic advances needed for the transition from dormant to operational for the many ECLSS components include procedure execution, planning, and in situ data analysis. Planning technology developments are covered in Section 3.13.3. The data analysis gaps to be filled could include methods of solving complex systems of differential dynamics equations using large amounts of sensor data in order to make real-time determinations about the status and health of the system. Though ground control will likely be in the loop during any transition or maintenance operations, data analysis for situations that could develop that may have time criticality constants less than the communication latency will be required. This technology is needed for many vehicle subsystems.

Similarly, autonomous control and scripting or procedure execution will be required to bring up the systems in a specific order, or to run prescribed maintenance checks on the equipment. This technology must be able to intelligently understand the status of the tasks being completed, whether each step has succeeded or failed, and everything in between. Though the ground controller in the loop could react in most cases to contingencies, it is important for this technology to understand when it is safe to move forward or when it is appropriate to wait for further instructions from the ground.

More advanced capabilities may also be required, such as task planning for event- or condition-based maintenance. Some examples include cross-system mitigations to control condensation or the use of robotic technology to aid in the recovery of a failed ECLSS component. Further discussion of task planning can be found in Section 3.13.3.

3.3.7 Criticality and Redundancy

The spacecraft life support system provides minimal capability during uninhabited mission phases. The required functions are limited to maintaining an environment in the habitat pressurized volume that is sufficient to support the vehicle systems (temperature and total pressure). It is critical that the ECLSS recover a habitable environment prior to crew return, and that all systems are capable of supporting the return transit phase.

The ECLSS must provide sufficient ventilation and thermal control to mitigate condensation. Excessive condensation could result in free water that could damage other vehicle systems. Free water on surfaces also promotes microbial growth which compromises the ability to recover a habitable environment. Free water may also be a result of leakage from the water system. Automatic detection and clean-up of free water is a challenge in not currently a capability provided by the ECLSS reference architecture. It is unclear how redundancy could be employed to mitigate free water.

An overpressure situation could compromise the habitat structure; failure of the relief valve could be immediate depending on the source of pressurization, making a redundant capability desirable in this case. The system should be designed to survive vacuum situations such that catastrophic failures will not occur due to under-pressurization.

Environmental monitoring must be maintained to provide insight into habitat and ECLSS health, and to provide alerts of anomalous or emergency conditions (e.g., fire). Environmental monitors should be redundant to maintain reliable insight into the environmental status, particularly in preparation for crew return, and to mitigate unnecessary consumables usage in responding to false alarms.

3.4 Guidance, Navigation, and Control

The GNC system manages the flight dynamics related state of the spacecraft during the course of dormant operations. The GNC system controls the translational state of the spacecraft, as well as its rotational orientation. The GNC system also controls the coordination of the spacecraft state during interaction with other vehicles, such as the Mars Ascent Vehicle or Orion crew vehicle conducting AR&D.

The GNC system uses a number of onboard sensors for state identification and management, such as inertial measurement units (IMUs), rate gyros, and star trackers. The GNC system commands effectors to control translational and rotational state, including orbit maintenance system jets, reaction control system (RCS) jets, and control moment gyros (CMGs).

3.4.1 Nominal Operations

The GNC system supports a number of spacecraft functions and objectives:

Orbital Maintenance
Collision avoidance
Robotics effects on state
Thermal control
Communication
Science operations

AR&D (during uncrewed activity)
Off-nominal conditions (effector issues, mission abort, etc.)

3.4.1.1 Navigation: Where am I?

During dormant operations, the spacecraft navigation system calculates state estimates (both translation and rotation) onboard the vehicle using available sensors. Due to sensor inaccuracies and drift, the navigation state must be occasionally updated to reduce navigation error. This cycle of calculation, drift, and update must be performed regularly during dormant operations.

3.4.1.2 Guidance: Where do I want to go?

In general, the commanded orbital trajectory managed by the guidance system is developed by flight operations in advance and the onboard system is designed to update and maintain the trajectory. For the dormant spacecraft, the trajectory will be calculated to minimize the need to perform correction maneuvers (to simplify operations and to minimize use of consumables).

3.4.1.3 Control: How do I get there?

The control system corrects for variations in vehicle state, and ensures that the spacecraft continues to operate safely and meet mission objectives. These mission objectives will drive control requirements, such as pointing accuracy or location above a certain point on an orbited object.

3.4.1.4 Autonomous Rendezvous and Docking

AR&D operations may be required during dormant spacecraft state, to support resupply by logistics modules before arrival of crew. It is also possible that modification of the vehicle (such as new modules or propulsion buses) will be accomplished without crew using AR&D. In cis-lunar space, the time delay is short enough that significant monitoring and support can be provided from the ground. As the distances from Earth grows and time delay increases, the requirements for AR&D will increase. At Mars, it is expected that any AR&D requirements must be performed without input from the ground.

The allocation of autonomous operations requirements will be different for the vehicles involved. The vehicle being docked with (called the “target”), is in a quiescent state and does not generally initiate motion for AR&D. The vehicle doing the docking (called the “chaser”) will have the autonomous GNC systems to initiate and execute the docking within constraints.

There are several requirements on AR&D activities: contact condition constraints to protect structure, plume impingement requirements to avoid contamination or unexpected dynamics, sensing of capture state to ensure proper alignment and final docking, etc.

The specification of relative sensors is key to successful operation of AR&D activities. Pre-placement of sensor elements on the habitat (such as reflectors) will be required.

3.4.2 Transition Operations

The dynamics state of the spacecraft must be monitored and maintained, in both dormant and crewed operations. Trajectories will be designed to minimize requirements for orbital maintenance and for use of expendables in either state. Rotational state commands may be driven by thermal, communications, or other mission-level requirements in either state. Rotational state may also be driven by science requirements, even when the habitat is uncrewed.

3.4.2.1 From Dormant to Active

The GNC system will be functioning during the dormancy state, so there will likely be minimal change in system state during this transition. If AR&D is a part of this transition, then the spacecraft attitude control requirements may change to accommodate the procedures, but from a system point of view there will be little change in performance.

The mass properties of the spacecraft stack will change when docking is complete. The control parameters will change to account for the change in mass state, but the effectors will be sized pre-mission to account for these changes. So once the control parameters have been adjusted, the attitude control and orbital maintenance capabilities will continue.

3.4.2.2 From Active to Dormant

When the crew has departed and AR&D support is complete, the GNC system will revert back to state maintenance, in compliance with the mission requirements described above. These functions will continue in the dormant state.

3.4.3 Contingency Operations

Contingency operations for GNC subsystems are divided into three broad categories: sensors and data input, effectors and vehicle control, and software issues. GNC can also support contingency operations having to do with other subsystem failures, and enable required mission re-planning such as mission aborts. Mission Planning often involves the Guidance and Targeting services of GNC, and in the future, some of these capabilities may be required onboard the vehicle for autonomous contingency response.

Like other domains, the GNC subsystem must operate safely and completely for crew safety and mission success. There is generally no dormant state. Contingency for GNC issues are generally managed either through hardware redundancy or through robustness or resilience to failure. For human spaceflight, safety requirements often demand redundancy.

GNC relies on sensors to develop state estimates for both attitude and position of the spacecraft. Contingency operations involve identification and response to problems within sensor data streams. Often software testing and system voting must be used to identify the faulty sensor, and to block the use of that data stream. In other cases, disparate sensors can be used as a cross-check (for instance, star tracker attitude compared to propagation of attitude state via rate gyro feedback). In order to improve the robustness of autonomous GNC sensor data, and therefore make GNC more robust to contingency, new onboard sensors are being evaluated that require less data from ground control. Optical navigation uses the size and location of known celestial objects to develop onboard state. X-ray navigation uses pulsar signals to triangulate location (although the sensor technology for this approach is still under development). In general, vehicle design must demonstrate the ability to collect state data for any contingency state, in order to safely execute vehicle state commands.

Contingency for GNC effector control is also managed through redundancy. Faults include a non-operational effector or a failed-on effector. In the case of RCS jets, GNC software will identify the faulty effector. Ideally this will be accomplished through direct sensor measurement such as chamber pressure transducers. Attempts have been made to identify faults through examination of dynamics, but this technique has proved complicated. It is recommended that direct sensor measurement be used. Once detected, the GNC software can account for the failure by selecting a new set of jets to perform actions (often this new set is less efficient than the original jet map). Other effector techniques such as CMGs require redundancy to deal with hardware failure. In the case of CMGs, it should be noted that the hardware can reach a state of saturation which requires other systems such as RCS to manage.

The GNC system can be used to support vehicle response to other contingency operations. For instance, if an issue with the ECLSS is identified and the duration of the mission had to be reduced (via Mission Abort), the GNC system would recalculate vehicle trajectories and would calculate a return to Earth. GNC would need to take into account the current known state of expendables such as fuel, and optimize based on either time or resources. Alternatively, if the fuel tank is leaking at an unacceptable rate, then the GNC system would also need to call an abort to provide an expendables margin to return.

3.4.4 Preventative Maintenance and Logistics Support

Historically, GNC systems tend to be embedded within the vehicle structure and difficult to R&R. Often crewed vehicles will depend on redundancy and hardware robustness. The vehicle can down-mode to the backup, and remain in a less-robust state since the options have been reduced. In the case of long-duration interplanetary missions, the ability to autonomously up-mode and recover the lost robustness is an important design consideration. Here autonomous is relative to ground control, and so includes both crew intervention and vehicle support.

Some effector systems such as RCS jets include valves and other moving elements. It is sometimes valuable to limit the duty-cycle (on/off cycles) of these components to maximize their mission life. It should be noted that these components are ground-tested to ensure significant margin. But in contingency operations, when unexpected loading can occur, it may be valuable to monitor and manage duty-cycling.

GNC sensors may include optical elements that need to be protected from wear. Issues such as scratching or exposure may impact performance and lifetime. Often preventative maintenance is achieved through careful mission planning (to avoid direct line to the sun, exposure to particles in the velocity vector, etc.).

For GNC systems, the issue of logistics is mainly one of preserving expendables. The amount of fuel may be fixed for the duration of the mission. Techniques are used to reduce these costs. For instance, when orbiting a central body, the vehicle will maintain a torque equilibrium attitude, that reduces the need for attitude correction. This attitude must be developed with many other vehicle requirements such as thermal control and power generation.

3.4.5 Autonomous Functions Needed

3.4.5.1 Nominal Operations

The capability to control the dynamics of a spacecraft to a destination in the solar system and return it safely to Earth is safety critical function that to date has relied primarily on support from the ground. This is due to several factors: 1) cost of validating autonomous GNC software to meet stringent safety requirements, 2) the capacity of onboard computing resources to execute autonomous GNC software, and 3) the type and reliability of onboard sensors and effectors to successfully execute the mission without ground support. These issues help identify the needs required by autonomous GNC functions.

Deep space vehicle missions have been able to rely on ground support for GNC functions because the time to criticality has been relatively high. The dynamics once in transit are relatively benign, and so the vehicle can operate safely for an extended period. However, the requirements change when considering human spaceflight, where the impact of an issue during any communication outage may be more severe. Additionally, when considering travel beyond Earth’s orbit, nominal spacecraft operations must deal with a number of communications issues. When traveling to Mars, the light-time-delay of communications can be on the order of tens of minutes, a relatively new constraint on human spaceflight. Some Mars missions may include “conjunctions” (orientations of spacecraft, Earth, sun and target) that could impact communication on the order of days or weeks. Finally, when the spacecraft arrives at the target destination, there may be requirements for AR&D, or for Entry, Descent, and Landing (EDL), where the time to criticality will be much shorter than can be supported by ground controllers on Earth. These requirements will demand an increase of onboard autonomy for GNC operations.

Two communication constraints will dictate the sufficient amount of Autonomous GNC on a spacecraft: 1) the nominal light-time communication delay constraint, and 2) the maximum credible loss of communication duration due to hardware failure on the vehicle.

3.4.5.1.1 Navigation

Navigation is the ability of the spacecraft to accurately calculate its position, either in absolute terms based to an inertial reference frame, or as a relative state when compared to another vehicle or body. The amount of acceptable position error in Navigation (a requirement on Autonomous GNC) is a function of the response to orbit maneuvering commands. The sensed state must be accurate enough for translation commands to succeed in meeting mission objectives (for instance, return to Earth).

Deep space vehicles have onboard sensors capable of monitoring and predicting vehicle state. However, these sensors to date have had accuracy issues such as drift, and over time the navigation error increases. After a defined time, the ground must upload a new state to correct this state drift.

The issues for Autonomous Navigation are focused more on hardware than software. The goal is to define a set of onboard sensors sufficient to maintain state, and with a bounded navigation error growth that ensures a translation command can be executed safely. In other words, the steady navigation state error using onboard sensors must be below a certain threshold. Concepts for Autonomous Navigation sensors include the following.

1) Optical Navigation: given a celestial body with a known location (an ephemeris), an atomic clock accurate enough to provide time, a good inertial attitude from a star tracker, and an optical device of known focal length, a vehicle can determine its location relative to the observed body and then by extension determine its inertial state, without help from the ground. This capability is only available when the celestial body is large enough in the visual field to support the calculations.

2) X-ray Navigation: given a set of pulsars distributed within the range of the vehicle sensors, and with an atomic clock for accurate timing, an interplanetary vehicle may be able to triangulate location from these pulsar signals and develop a navigation state independent of the ground.

3) New IMU technology: sensor developers are working on new technologies (such as cold-atom IMUs), to reduce the drift rate of sensor error. While not supporting complete onboard autonomy as the other techniques may, improving sensors and reducing drift rates will increase the minimum time to criticality and therefore increase mission safety.

3.4.5.1.2 Guidance and Targeting

Targeting is the process of selecting a target state for a spacecraft, and then developing a plan for achieving that state usually through the application of propulsive force (either discretely or continuous over time). Guidance is the ability for the spacecraft to maintain the desired state during a mission, and may require mid-course or orbit maintenance corrections due to errors in force application or to environmental effects. Targeting is usually performed on the ground, first by mission planners months or years before a mission, and later during a mission itself to deal with both nominal and off-nominal (or abort) conditions. Targets and burn plans are uploaded to a vehicle periodically as needed. Guidance is normally run on board the vehicle, but often with bounds of operation. If a position error exceeds a threshold, retargeting may be required.

To date, the time to criticality has not required significant onboard targeting capability. Once a stable interplanetary orbit has been developed, targeting can be performed on the ground and uploaded as needed.

Two deep space mission requirements impact this assumption. First, if rendezvous with a celestial body is required that occurs many light-minutes from Earth, the dynamics of relative motion will likely make control from the ground impossible. So the spacecraft must use autonomous targeting and guidance to perform this activity. Second, if there is an extended loss-of-communication (either planned due to celestial orientation with Earth, or as a result of failure), the vehicle must be capable of controlling its translational state sufficient to provide crew safety as well as enable mission success.

It should be noted that during a multi-year mission far from Earth, abort opportunities are very limited. The propulsive force required to modify trajectories arbitrarily along a mission path to get back to Earth is not available. The goal of Autonomous G&T is to position the spacecraft in a state of extended safe operation, whether that is a transit trajectory to another planet or a safe parking orbit around a planet. The worst case scenario from a safety point of view is the burn to return to Earth. This timing is critical because of the required planetary alignment for safe return. If the return window is missed, it could be months or years before the correct alignment is reached again. For the narrow case of a loss-of-communication near this time, the TTC is drastically reduced. This problem can be dealt with two ways: 1) a pre-defined set of burns is planned and executed in case of loss-of-communication (this solution assumes that the states of the vehicle and the planets is predictable enough that the pre-defined plan will succeed with no real-time state information); and 2) the vehicle would use real-time state information (which may include a degraded capability due to hardware failure), and would re-target the appropriate trajectory for safe return. Onboard targeting would be required in this scenario.

GNC engineers have been considering how to reduce the computational requirements for developing onboard targeting solutions. Often the solution requires iteration and optimization algorithms to account for constraints such as consumables (fuel, oxygen) and time to target. If the vehicle has sufficiently accurate state information from Autonomous Navigation, and if the targeting algorithm can get required constraints from the other vehicle systems, then Autonomous Targeting could calculate a new trajectory and provide a burn plan for the vehicle to execute. Of course, this is an incredibly dangerous and safety-critical maneuver. The only reason a project would attempt this type of capability is 1) if the loss-of-communication scenario is credible enough to force onboard targeting, and 2) if the solution to use a pre-canned burn plan is not sufficient to ensure safe return.

3.4.5.1.3 Control

The GNC vehicle control software is normally onboard the vehicle, and not supplied by the ground. This software is usually automatic (prescribed and validated by the ground), and occasionally it can be adaptive (taking real-time information and adjusting control commands). Rarely is control “autonomous” in the sense that the system develops new ways of executing control. Effector failure (such as a jet blockage) is usually designed in pre-mission, and alternative control commands are created to bypass the failure.

Historically, translational state is controlled using a “bang-bang” system of discrete burns to correct error or change orbit. With the use of Solar Electric Propulsion and persistent low-acceleration force application, future autonomous control algorithms may have different requirements for maintaining continuous control of these active effectors. This could imply more onboard capability for state management independent of ground control (compared to the uplinking and execution of discrete burns, calculated on the ground).

3.4.5.1.4 Interaction with Autonomy Frameworks

In many cases, the implementation of Autonomous GNC may be realized completely within the GNC system itself. Solutions such as X-ray Navigation support autonomous operations but do not necessarily involve an Autonomous Framework. However, when issues with other systems dictate a change in vehicle state controlled by GNC, then there must be a clear interface between the two systems. In addition, displays and controls are used to ensure onboard crew is aware and actively participating in critical GNC decisions.

For example, if ECLSS detects a cabin leak that is sufficiently large, then the autonomous system may request the GNC system to conduct an orbit burn earlier (to get the crew back faster than planned). Any of these responses will be a function of the mission phase, consumables and other vehicle properties, etc.

3.4.5.2 Contingency Operations

The GNC system must be robust and resilient to system failures during a mission. The level of robustness or redundancy is a function of system criticality, as described in Section 3.1. Again, the level of engagement with autonomy may be slight.

3.4.5.2.1 System Redundancy for Continued GNC Performance

If the off-nominal condition is purely a hardware failure, not requiring a mission replan, then the GNC system will automatically switch to backup systems to support mission execution. For instance, it may be required to fly many star trackers, to ensure GNC has a good inertial attitude state. Alternatively, one star tracker rated to a higher level of robustness may be flown.

3.4.5.2.2 Mission Re-planning and Abort

It is possible that a system failure or a decision from humans requires a mission replan. In that case, if the vehicle is out of communication with the ground, the crew will work with the Autonomous GNC capability to identify a safe credible plan. Accommodations must be made to support onboard, real-time verification of new (novel) plans before they are implemented.

In very limited cases, an abort must be performed without intervention from ground or even crew. For instance during EDL activities, the response time (time to criticality) may be on the order of seconds. For these cases, the GNC is designed to handle the failure. Note that this is may not be “autonomous” in the sense of on-the-fly decision-making, but will likely be a pre-design control mode used by GNC to account for the error.

3.4.6 Development Plan

The developments and autonomy needs for the Guidance, Navigation and Control subsystem largely center around operations that occur with humans on board, such as targeting, deep space navigation, or EDL. However, there are autonomous functions needed for uncrewed mission phases as well. The developments needed can be described by both system engineering needs as well as algorithmic needs.

3.4.6.1 Systems Engineering for GNC Autonomy

Sensors for accurate navigation are the most pressing concern for autonomy while uncrewed. Sensor packages must be developed to ensure that the stack up of errors is not too great over a period of time sufficient for the amount of communication blackout that will be experienced. The uncrewed time is not the forcing function for the design of these sensors, however, so developments in this area should be based upon the needs of the crew transit phases of the mission.

A study should be commissioned to understand the operational concepts required for targeting for human deep space missions. For example, using pre-planned trajectories and burn plans may be sufficiently accurate for emergency returns from Mars orbit, given that these are updated by the ground on a somewhat regular basis. A trade between the robustness of the communications system versus the needed investment into on board autonomous targeting would determine where development efforts should be focused given that pre-planned trajectories are found to be insufficient. Understanding the integration and contributions of all parts of the spacecraft to the problem solved by a particular subsystem is a systems engineering practice that will be essential to developing the autonomous functions that are needed for this design reference mission.

The architecture of the GNC system, including which sensors are used, back-ups for these sensors, and the robustness of sensors, should be well-understood given the requirements of the system. The system requirements are likely to be driven by crewed mission phases, but investment into the effect that autonomy places on the sensors in the GNC system is important to make.

3.4.6.2 Autonomous Algorithms for GNC

The need for autonomous algorithm development will be highly dependent on the desired locus of authority between the ground controllers and the spacecraft, given several other parameters of the mission, such as expected communications availability during critical junctions, the requirement to autonomous rendezvous and dock, or the need for EDL. However, some algorithmic investment is warranted.

3.4.6.2.1 Propagation of Error

Algorithms to accurately predict the propagation of error given sensor models, and to adapt to changing conditions for these sensors, would be useful for autonomous navigation. These could be used in conjunction with improving the accuracy of the sensors. Algorithms could be developed that take advantage of the different properties of sensors used for determining position and orientation in space in order to find a solution, autonomously, that has the proper accuracy.

3.4.6.2.2 Mission Re-planning

Limited scenarios may drive the need for autonomous mission re-planning. While planning is a technology needed by many subsystems (and is discussed in Section 3.13.3), the criticality of these plans drives the need for in situ verification of the plan before it is executed. The complexity of these plans will drive both the algorithm used to create the plan and the verification method used to ensure the safety and correctness of the plan.

3.4.6.2.3 Autonomous Reactions to Emergency Scenarios

Control of the spacecraft is already achieved on board the spacecraft, but there may be need to adjust the control automatically in the case of certain emergencies, such as a rapid depress. The controller parameters or mode may need to change based on the new dynamics of the spacecraft. Interaction with a vehicle system manager (VSM) would be necessary to reconfigure the GNC system in the recovery from an emergency, but the GNC system must be designed to interface with the VSM for a successful integrated system.

3.4.7 Criticality and Redundancy

The operation of GNC is critical to the safe and successful execution of a space mission. Guidance is required to ensure the spacecraft arrives at the desired location, within time and consumables constraints. Navigation is required to provide state information to crew and ground, and from which to predict future desired states. Control is required to actually perform state changes based on mission requirements. All these functions are safety critical operations with crew, and are mission critical with or without crew. Some of these critical requirements are allocated or assumed for other vehicle systems as well (sufficient fuel to change state, sufficient power for GNC systems, sufficient software capability to run GNC applications).

Human-critical spacecraft use redundancy to protect GNC critical functions from failing. There is hardware redundancy, across a number of systems. There may be duplication of similar sensors to overcome unexpected hardware failure. There may also be the use of dissimilar sensors, to provide robustness and also to protect from various space conditions such as lighting. For instance, a spacecraft may use both a camera and an infra-red sensor to identify other vehicles when controlling relative motion. GNC may leverage hardware redundancy in other systems, for instance utilizing different RCS jets if one jet fails. Normally the GNC has a “jet map” within the algorithms to both identify the failure, and quickly select the best alternative.

GNC may also depend on voting. When using redundant sensors, there are often a number of sensor values. The software must “vote” to ensure that the correct value is used. If one sensor fails, the other sensors will “vote” it out. This implies that there must be more than two sensors operating. Should one of two fail, there is no chance of knowing which is correct.

GNC takes advantage of avionics redundancy to ensure that the loss of one computer does not prevent GNC operation. The number of failures to protect for is a function of the cost of the mission, and whether crew is present. GNC criticality is a function of mission phase. During highly dynamic mission phases, such as ascent, entry, or orbit transition, GNC depends on system robustness. If a processor fails, there is often no time to correct the problem so the system must have enough capacity to continue even with the failure. For other mission phases, such as orbit or transit, the system may operate for extended periods without command. During these phases, it may be possible to correct problems in GNC hardware or software.

As with other software products, GNC software for human spacecraft is considered Class A/Safety Critical. This designation affects not only the level of testing performed, but also the level of cross-check and software assurance required. One issue is the possibility of a common catastrophic software error within the GNC code that could not be identified through simple voting or other resiliency techniques. Since GNC is required for the safe return to earth, some projects have elected to create backup flight software, which has some level of independence from the original code. The level of independence is a trade of risk and cost. For the Space Shuttle, a completely independent set of flight software was specified, designed, built, and tested. There was no commonality with the original GNC flight software (FSW), thereby providing a level of protection against the common-source error. Other options include using the same GNC code, but running the code on a different processor and using a different code compiler. This approach helps protect against compiler or hardware sources for software error, but does not protect for design flaws. One open question is whether new, better software testing techniques can more completely minimize the risk of a major design flaw in the source code. The more independent and complete the backup system, the more cost and vehicle complexity is added (which brings its own set of risks).

3.5 Power

Power system operations can be broken down into three categories: battery maintenance, solar array operations, and general load operations.

3.5.1 Nominal Operations

Battery maintenance involves maintaining the batteries at proper state-of-charge (voltage) for long term life. Based on the battery design, this is probably not 100% charge. Additionally, periodic (yearly) battery health checks (load and charge/discharge cycle) will need to be performed. Battery temperature must be maintained for optimum long term life. Eclipse or day/night operations involve pre-charging the battery to last through eclipse or night, load shedding for eclipse or night and/or pre-warming subsystems for eclipse or night to reduce electric power usage during eclipse or night.

Solar array operations involve maintaining optimum orientation for power production and self-protection. Solar arrays require periodic dust mitigation for surface operations as well as involvement in the periodic (yearly) battery health check (load and charge/discharge cycle). Solar arrays are also essential to maintaining battery temperature for optimum long term life.

General load operations include phasing occasional loads to avoid peak power issue, implementing partial power-downs, and managing system power up sequence from deep fault conditions (Lazarus mode).

3.5.2 Transition Operations

3.5.2.1 Transitioning from Full Operations to Dormancy

Lower overall power requirements in dormancy may allow solar arrays to be reconfigured to make them age more slowly. For instance, some sections of the arrays may be tilted to reduce their operating temperature, or reduce exposure to micrometeors, solar ultraviolet, or other environmental factors. Lower overall power requirements in dormancy will mean that the batteries do not have to store as much energy for eclipse or night side operations. This allows the batteries to be operated over a range of state-of-charge optimized to extend battery life. Finally, the electrical power system can certainly power down individual loads in subsystems, but that is more a function of the subsystems themselves.

3.5.2.2 Transitioning from Dormancy to Full Operations

In order to meet the increased electrical power needs of full operation, the electric power system will reorient the solar arrays to maximize power production.

In order to prepare for full operations, the batteries will be fully charged before the start of full operations. During full operations, they will be operated over a wider state-of-charge range than during dormancy.

3.5.3 Contingency Operations

3.5.3.1 Solar arrays

In general, solar arrays must maintain a certain orientation with respect to the sun in order to generate the desired power level. It is often necessary to have the angle of the arrays with respect to the vehicle change so that the arrays can track the sun. There are many contingency scenarios in which the arrays lose alignment with the sun, or are not able to track the sun. In the event that the arrays can still track, the tracking mechanism must be able to reorient the arrays before the vehicle electrical energy storage (batteries) are depleted.

In the event that the array orientation mechanisms are not fully functional, the remaining functionality must be used in the optimum way to generate the most energy. This may include the use of systems not intended for array orientation, such as reaction wheels or reaction jets.

Array orientation may play a role in the response to contingencies in other subsystems. For instance, the array orientation may be changed to shade or not to shade parts of the structure in order to achieve thermal effects.

Solar arrays sized to meet the power requirements over the entire mission will have excess capacity in many phases of the mission. This is generally managed by electrically shunting around preplanned sections of the array. In the event of damage to part of the array, this shunting function must be reconfigured so that the shunted sections are the damaged sections.

3.5.3.2 Batteries

Batteries have internal failure modes that may require activation of contingency hazard controls. For instance, large terrestrial battery systems often have fire suppression system.

The batteries may have extra components (redundant cells) designed so that the battery can reconfigure around nonfunctional components such as cells that have shorted or dried out (seal failure).

3.5.3.3 Power Management and Distribution

The PMAD subsystem (sometimes known by other phrases, such as Electric Power Distribution and Control) transfers electrical power from the generation and storage subsystems to the various loads.

One function of the PMAD subsystem is fault isolation. If a load on a power bus begins to draw excessive power due to an internal fault, there are two reasons to isolate it. The excess power may be causing some sort of damage in the load, and the excess power draw interferes with the ability of other loads to draw the power that they need. The PMAD system must detect this problem and disconnect the problematic load from the bus.

PMAD subsystems are often configured into more than one power bus. This allows higher level fault isolation, and critical loads often take power from more than one bus in order to assure functionality if one bus malfunctions.

The PMAD subsystem often plays a role in determining the cause of anomalies in other subsystems. Such subsystems often have components that use electrical power, and which have characteristic signatures of power usage. The PMAD subsystem has instrumentation that allows such signatures to be measured.

The PMAD subsystem also plays a role in contingencies of other subsystems by allowing sections of the subsystems to be placed in known nonoperational states by cutting off power to them.

3.5.4 Preventative Maintenance and Logistics Support

Solar arrays are generally designed to last the whole mission without need for preventive maintenance. Since arrays can have moving mechanical parts like gimbals or deployment mechanisms, it is possible that such mechanisms would need maintenance like lubrication.

Solar arrays are not designed to need logistical resupply, with the possible exception of modular arrays designed for field replacement of modular sections.

Battery systems are designed to last the whole mission without a need for preventive maintenance. Aerospace battery systems do not have consumable components that would generate logistical concerns. If the battery system is designed with several field replaceable units, one or more spare units would require logistics support.

PMAD systems generally have a large number of identical power switch/current limiter modules each of which connects a load to a power bus. Experience shows that in large systems a few of these modules may fail, either due to internal faults, or due to load problems. One approach is to use in-place redundant switches on critical loads, but another approach is to have field replaceable switches or switch modules. If the total number of switches and critical loads is large enough, and replacement is feasible, field replaceable modules may require less total mass than in-place redundant switches.

If the PMAD system is designed with field replaceable modular units, storing spare units would be a logistics issue. Unlike with batteries and solar arrays, PMAD systems with field replaceable subassemblies are actively being designed at NASA, so including replacement modules in logistical planning would be reasonable.

3.5.5 Autonomous Functions Needed

There are several technology gaps that can be developed in order to make a more robust and autonomous electrical power system for complex human spacecraft. They are detailed in the subsections that follow.

3.5.5.1 Autonomous Contingency Management

The ability to determine the need for shunting failed components (like solar panels) and executing the necessary procedure is required in cases where the TTC would be too short for ground controllers to address. Likewise, a fire suppression or battery protection method to allow autonomous battery operation is required.

3.5.5.2 Optimal Load Shedding

Electrical power sources may degrade rapidly, faster than communication bandwidth would allow an optimum response initiated by ground controllers. The state-of-the-art is to have an automatic load shed that is safe but may be suboptimal. Technology is needed to be capable of identifying need to load shed, for example, due to the degradation of power source or power storage. Also, an algorithm to identify an optimal load shedding plan automatically is required.

3.5.5.3 Smart Telemetry Management

Communication bandwidth to provide continuous monitoring at a sample rate sufficient to show the details necessary to interpret anomalies or "out-of-family" events is not available for the ground controllers on Earth. As such, it is important to create an autonomous system to record, keep, and send to Earth high rate data important to the anomaly or event. Algorithms are needed to always identify when these significant events, anomalies, or "out-of-family" waveforms occur. Then, instrumentation must be present to record the relevant data, and then the communication of this information to a higher level expert system or ground control is required.

This problem is difficult in several ways. First, for the mass- and bandwidth-constrained spacecraft, the hardware needed to sense, store, and transmit events must be carefully considered. Second, the determination of a good general definition for out-of-family may be difficult, as changes may be subtle or wholly unexpected. Once the event has been captured, using the knowledge in the best way for the subsystem with the problem is a third challenge.

3.5.5.4 Optimum Battery Management

Maximizing battery lifetime and performance is best served by using different battery charge and cycling set points or algorithms. Changing the set points by ground control uses communications bandwidth and relies on communication being available. An autonomous system to manage the batteries for optimal performance and lifetime would reduce ground operator workload and potentially prolong battery utility. Challenges include identifying the need to change configuration (for example, identifying mission phase) and ensuring that the autonomy would reliably fulfill this critical function.

3.5.5.5 Prognosis

Present fault detection misses the chance to detect initial small electrical faults before they propagate into large, damaging, unambiguous faults. Developments needed include autonomous recognition of subtle signatures of incipient faults such as series arcing in wiring and the enhancement of present current measurements to allow detection of ground fault currents while respecting mass constraints.

3.5.5.6 Optimum Solar Array Orientation Control Management

Failures in solar array orientation control mechanisms may be worked around by reaction control wheels or jets, but present systems require ground intervention to do this. This could be achieved by developing control algorithms that recognize the fault and change attitude control logic to compensate. However, this is complicated to do without causing undesirable unintended effects. Much analysis and simulation would be necessary.

3.5.6 Development Plan

Autonomous system development for electrical power systems has two focuses. First, the system design to optimize the data that would be provided to the system (while respecting robustness, mass, and power constraints) is important to be completed with the autonomous functions in mind. Second, diagnosis, prognosis, and planning capabilities must be advanced. While developments may leverage various industrial and university efforts that are ongoing, but particular attention will be paid to the needed advancements that are largely limited to space applications.

3.5.6.1 Systems Engineering Design for Electrical Power Systems

Many of the gaps identified in the corresponding autonomy needs section require the system to make decisions about the state of the system given the data that is returned. Given that possible faults are largely known, the difficulty lies within determining the specific fault mode given the signature that can be measured. As such, it is important to design the sensor network for the autonomous capabilities that will be required of the system. This integrated design of the system in conjunction with the autonomous functionalities will ensure sufficient observation matrices in order to achieve overall system success, while respecting the cost, mass, and power constraints in the design. A more detailed treatment of sensor networks can be found in Section 3.13.1. The complexity of the overall system will be reduced with careful design and thought given to these functionalities during even the requirements generation and early design stages.

3.5.6.2 Autonomous Electrical System Capabilities

3.5.6.2.1 Fault Detection and System Health Management

Reacting to or recording anomalies requires an accurate determination of the fault or failure. Detection of faults, either via using a fault model or without, is an important part of filling the autonomy gaps. While this technology is required for many subsystems and is covered more fully in Section 3.13.2, these functions will likely need to be able to execute in a somewhat distributed manner, with subsystems making decisions as able before a centralized system gets involved. Development on effective ways to deploy these distributed models is needed.

3.5.6.2.2 Prognosis and Sensor Advancements

Model invalidation over large time scales using slowly changing data denoting incipient faults is a core problem for an autonomous electrical power system. Learning technologies that are able to define accurate, specific models for each part of the system could characterize individual elements so that even small discrepancies would be noticed. Decisions could be fairly trivial, such as report irregularities to ground control, but the real advancement would be to enable the prognosis capability on spacecraft systems with mass and computation constraints. Sensor advancements that reduce the footprint (volume, mass, cost, power) of the device could enable new autonomous prognosis capabilities by virtue of more or better data.

3.5.6.2.3 Verification and Validation of Autonomous Systems

Given that the electrical power system is a critical one during both crewed and uncrewed periods, it is essential that any autonomous functionality be sufficiently tested and guaranteed before deployment. This technology is outlined more fully in Section 3.13.4.

3.5.6.2.4 Task Planning and Optimization

Because the control and configuration of electrical power systems affect the spacecraft so profoundly, the ability to determine control actions to reconfigure and optimize the system is a complex and critical task. Constrained task planning that involves solving complex dynamical systems to find a provably correct and optimal plan is required to bridge this autonomy gap. While this technology is likely to apply to systems on Earth as well, NASA investment is required due to the very low TRL of this capability. This technology is outlined more fully in Section 3.13.3.

3.5.7 Criticality and Redundancy

The primary electrical power generation system is necessary to ensure the health of the vehicle. Almost every other system (with the possible exception of structure) uses electrical power in important ways. Spacecraft are often built with a second or third electrical bus complete with its own primary power source, so it is 1R3 criticality.

Not all the loads on the electrical power system are critical, and it is possible for the system to respond to degradations of its capacity by shutting off less critical loads.

The electrical power generation system has failure modes that involve immediate loss of function, but it can also suffer long term degradation of function. For instance, solar arrays age with time and radiation exposure. It is not always possible to anticipate the rate of degradation. If the rate of degradation is greater than anticipated, the result will be a slow reduction of function. Arrays are designed with additional cells to build in margin for this possibility.

The most likely primary power source is solar arrays. In use, arrays can show planned intermittence (eclipses or reconfiguration for operations like dockings) and unplanned intermittences. Because of the planned intermittencies, the electrical power system has energy storage, either batteries or fuel cell/electrolyzers. These are sized to store enough energy to cover intermittencies. Gaps of less than a day are usually handled with batteries.

Since the batteries store significant amounts of energy, they present a significant hazard, and have failure modes that are more significant than just fail to function, such as explosion or fire. This means in operational phases in which battery redundancy is not needed (long stretches of operation with no crew and no eclipses), it may make sense to place some batteries in a low charge state to reduce the stored energy and hence the effect of a catastrophic internal short in the battery. The highest performance cell chemistry (lithium ion) also ages more slowly when stored in this state.

Batteries often have redundant cells to handle long term degradation such as calendar and cycle life. Since the cells are usually in close physical and electrical proximity, there is redundancy on a battery level, too. Tying in with the whole multiple electrical bus idea, the batteries show the same level of redundancy.

The PMAD subsystem transfers electrical energy from the arrays and batteries to the loads. As discussed, there are often one or more power busses to achieve redundancy.

The PMAD system has an important role to play in isolating electrical failures and preventing the propagation of failures from one load to others (circuit breaker and fuse type current limiting). This function is usually not redundant on a per-load level, but there can be a hierarchy of current limiting devices that would limit extensive failure propagation. One side effect of the fault isolation function is the ability to reconfigure loads to achieve subsystem goals.

The PMAD system can play a role in inhibiting functions (like pyro devices and propulsion) where inappropriate activation would be hazardous.

3.6 Propulsion

The baseline assumption for this propulsion and power bus (PPB) is an electric propulsion system powered by solar arrays. The ACS is assumed to be hydrazine. However, other options for the PPB could be a nuclear propulsion system with an oxygen-based attitude control system.

3.6.1 Nominal Operations

The propulsion system will be partially active during nominal dormancy periods for a spacecraft in Mars orbit. The PPB system will be used to maintain attitude as required for communications, thermal, and power. The propulsion system will respond to GNC commands to maintain attitude as required.

Assuming there is no orbital decay due to drag in Mars orbit, the solar electric system or nuclear systems will most likely not be used for delta-V maneuvers during the dormant periods. There may be a requirement for delta-V maneuver near the end of the dormant time periods. In terms of specific operations during dormant periods, there are requirements to monitor propellants (such as xenon for electric propulsion) for leakage.

The ACS will be active and used periodically during the dormancy. These operations include monitoring temperature and pressure of the propellants, monitor system and engine temperatures to detect leakage, configuring and warm up cat bed heaters for ACS burns as required, monitor chamber pressure or GNC rates for engine operational health check. Leakage of propellant is a key check that may be done by measuring a decay in temperature or mass. Mass may be directly measured via a gauge or by pressure-volume-temperature (PVT) calculation via an algorithm.

Examples of faults are leakage, failed open or closed valves, and a failed ACS engine (heater, temp measurement, pressure measurement, catalyst bed, etc.). These faults typically require a reconfiguration of valves to isolate the leakage, or the activation of or switch to redundant end effectors.

3.6.2 Transition Operations

This describes how the propulsion system will go functionally from a state where humans are in the spacecraft to a state where humans are not in the spacecraft and back again. There are several assumptions. Nominally, the propulsion system will be in an operating state at all times, whether crew tended or not. The system may not be operating at full capability depending on any faults that may have occurred prior to the transition. These faults may require some contingency operations (repair), logistics, or maintenance that occur during transition in order for the system to be safed. The propulsion system shall respond to GNC commands to maintain attitude and trajectory as required.

Another consideration is the possible need for propulsion during AR&D operations when the spacecraft is not crewed for the purposes of resupply.

Alternatively, the crewed vehicle mission might include a logistics element that is co-manifested, which removes the need for completely AR&D since the crew is involved. Both scenarios are considered for now.

3.6.2.1 From Dormant to Active

The propulsion system will be functioning during the dormancy stage, so there will likely be minimal change in system state during this transition. The propulsion system will likely activate pressurization systems if not already active. If checkouts are part of transition, then these may be scheduled at this time, however since propulsion is active during dormancy, checkout can be done prior to transition. If AR&D is a part of this transition, such as with a logistics module, then the spacecraft attitude control requirements may change to accommodate the procedures, but from a system point of view there will be little change in performance.

The mass properties of the spacecraft stack will change when docking is complete (for a logistics mission or a crewed mission). The control parameters will change to account for the change in mass state, but the effectors will be sized pre-mission to account for these changes. Once the control parameters have been adjusted, the attitude control and orbital maintenance capabilities will continue.

3.6.2.2 From Active to Dormant

When the crew or logistics module has departed and AR&D support is complete, the propulsion system will continue in the dormant state, however with expected changes in commanding of thrusters or other effectors.

3.6.3 Contingency Operations

The propulsion system may be operational during all phases of the mission whether dormant or active. Therefore, contingency actions may be required at any time. There will some form of automated fault detection isolation and recovery procedures that will running at all times at some interval that is determined by insuring that there is an amount of time available to recover. The contingency operations are broken down in the critical functions being provided by the propulsion system.

1) **Provide attitude thrust**. Thrusters are fired autonomously per a table onboard and fault flags. Contingency operations may be to specify selection order for life, performance, other degraded performance reasons, such as leakage detected or chamber pressure issues. Jet failures, as detected by GNC rate issues, chamber pressure, or temperature, may require immediate switch to a redundant jet.
2) **Provide translation thrust**. Translation thrusters are fired per a GNC burn plan. If an engine fault is detected by GNC rate issues, chamber pressure, or temperature, then other thrusters will be selected. This may require changes to burn durations (GNC determines down moding) that need to be made immediately.
3) **Valve Configuration for pressurization of the propellants tanks.** There are several possible states for these valves: closed, open, or actively controlled open/closed. For example, the pressurization valves may be closed between translation burns. Thus, the attitude control systems may operate in blowdown between translation burns. Telemetry is used to check valve states and monitor pressure for indications of leakage. Valve full open failures and relief valve rupture will require immediate response to close the isolation valve. If an isolation valve or regulator leakage is detected by tank pressure increasing above a limit, then the pressurization control valve or regulator leakage may require the isolation valve to be kept closed. Contingency operations may be to inhibit

specific isolation valve operation. If failure of the nominal pressurization system is detected through pressure measurements, then the redundant or cross-feed pressurization system may be selected.

4) **Propellant storage and line temperature.** The set points of the propulsion TCS may be adjusted. The system will monitor temperatures for failures. Failure of heaters may require adjusted selection of strings or change in attitude. The time to effect may be on the order of minutes to hours.
5) **Provide quantity of pressurant and propellant remaining.** This is used for planning burns and assessing health of tanks, lines, and engines. Insufficient remaining propellant may require changes to the mission. There are several techniques to gauge the propellant remaining, such as pressure-volume-temperature calculations, burn time integrations, flow meters, and direct propellant quantity measurements such as capacitance probes or radio-frequency mass gauges. The propellant mass gauging technique may involve looking at telemetry and performing ground or onboard calculations to allow a more rapid response to problems in the quantity of propellant. This is used for assessing health of tanks, lines, and engines and estimating propellant remaining. If rapid decay or propellant quantity is detected, such as due to a leak, then a rapid response is required. The response would be to determine where the leak is and to isolate by closing a valve. This may need to be responded to autonomously onboard. This is not typically done for current spacecraft.
6) **Detect tank, valve, or engine leakage.** Detecting tank, valve, or engine leaks will use techniques other than mass gauging. This will involve temperature or manifold pressure measurements to detect the leak. These direct leak detection techniques will be compared to what the mass quantity gauges are indicating. Detected leaks may require rapid response in order to isolate the component/system if rapid leak detected. The catastrophic hazards are depletion of propellant and engine detonation.

3.6.4 Preventative Maintenance and Logistics Support

Typically the spacecraft propulsion systems are qualified to not require any maintenance or repair over the expected life and cycles. This will mean a minimum of single fault tolerant system, as failures historically are possible for long-duration spacecraft. Some mission architectures do require resupply of propellants and pressurants. A human rated propulsion system may have the following maintenance and logistics operations performed.

3.6.4.1 Maintenance

Operation of system for purposes of check-out tests may be required if the function has not been demonstrated recently. This may be tied to transition operations. Hypergolic systems require periodic thruster firings to prevent accumulation of iron nitrates and heater operation to remove any condensed ice or propellants. Cryogenic thrusters may require heating or operation of the ignition device to remove any condensed moisture or propellant ice. Electric propulsion systems are non-toxic so replacement of thrusters at end of life is possible. The location (i.e., in-space, planetary surface) of the propulsion system may drive certain maintenance operations such as purges, pad pressures, covers. Some ancillary systems that are not connected to propellant may require maintenance such as cryocoolers, electric pumps and batteries, and compressors.

3.6.4.2 Logistics

Several logistics task may be required. First, the system must plan for propellant and GHe re-supply if required. Keeping track of the quantities of these resources is an important part of the plan.

If required, the system must conduct a propellant and GHe resupply by making and verifying Fluid connection (part of docking or just after docking) and preparing the system to receive fluid (may require receiver tank venting). Next, the transfer propellants of pressurants requires operating compressor or pump and measuring the quantity transferred at least two different ways. Finally, the system must disconnect fluids and verify no leaks.

Another task is keeping track of life cycles for tanks, valve, thrusters. Adjustments to command tables (sometimes in GNC) for items like thruster selection, manifold or valve selection, and tank selection may be required.

Systems are not typically designed for repair or routine replacement of parts, although use of cryogenic non-toxic propellants could facilitate this. Even hypergolic propulsions systems could be repaired if capability to inert the system is designed into the system.

3.6.5 Autonomous Functions Needed

Several technology gaps exist for the autonomous operation of the propulsion system.

One gap is the automation of external leak detection isolation and recovery for tanks, lines, components, and thrusters. The time to criticality depends on the size of the leak, but can be as short as seconds. This is usually done manually on the ground. The need is to develop sensor system designs and algorithms that autonomously monitor temperatures, pressures, and propellant quantities to determine if a leak exist, then isolate the leak. If it is a thruster, then thrusters on that manifold may be isolated and removed from the jet select table. GNC may be required to take further action for control.

Leak detection is complex, requiring many sensors. The possibility of false readings exists. Determining where the leak is occurring may require a process of isolating a manifold and checking if that fixes the leak. Since there are other fluids from other subsystems typically nearby, some form of centralized intelligence may be required to determine which system is leaking. The process may also require computing a leak rate based on pressure decay measurements.

Another gap is the full open failure of a pressurization valve or regulator which will cause pressure relief devices to open and vent GHe externally and therefore consume GHe at a higher rate than nominal. This can cause rapid depletion of GHe or propellant. One method to detect this is helium mass quantity calculations based on pressures, temperatures, and possibly sensors at the relief device. An algorithm will need to be developed to isolate and recover from the failed pressurization regulator.

Another gap is maintenance and logistics. Maintenance may have some automation needs. This may be to periodically fire a thruster to prevent accumulation of substances that cause engine flow decay or leaks. This will change jet priorities in the jet select table. Periodic valve, mechanism, or motor operation may require operation to prevent sticking or binding. Maintenance will need to be performed autonomously, but can be scheduled by the ground.

Logistics, such as refueling of the spacecraft, will also need to be performed autonomously. This will involve all spacecraft functions for AR&D, mechanisms for attaching refueling couplings, control of the propellant transfer, etc. Logistics such as refueling will require centralized intelligence with subsystem automation since this will involve multiple systems (GNC, thermal, power, propulsion). Refueling is a complex operation with many of the same functions described above for pressure control, leak detections, quantity gauging, and flow control.

3.6.6 Development Plan

The development plan for technologies that will enable autonomous operation of the propulsion system as needed above and beyond current automation for this design reference mission can be grouped into two categories. First, the design of the propulsion system is an important factor into being able to control and maintain the system autonomously. Second, algorithms are required to make sense of data and mitigate any failures or perform any maintenance that may be needed.

3.6.6.1 System Engineering for Propulsion Autonomy

Due to the difficulty in pin-pointing leaks in the propulsion system, the overall design for the autonomous solution will be some combination of sensors, system design, and autonomous algorithms. Sensor network design is treated in Section 3.13.1. While fully instrumenting each line, tank, component, and thruster may be cost prohibitive in terms of mass, power, and processing, observations of each of these things must be possible to provide a solution to the health management problem. As such, investment should be made both into the design of sensors that could provide insight into leaks and failures as well as into system designs that allow small perturbations to the system to provide extra information. For example, sensors could be placed in lines such that some combination of valve closings could provide the information needed to pinpoint the point of the leak. This would reduce the number of sensors needed at the cost of the development of active diagnosis methods to deduce the location of the failure. The complexity of the observation algorithm, however is highly dependent on system design. Investment into the combination of system design and autonomous algorithms for system health management should be prioritized.

The potential need for autonomous refueling also places requirements onto the propulsion system from a system design standpoint. The spacecraft and the propulsion subsystem must be designed to accommodate autonomous refueling, for example, with aids for AR&D, with interfaces that allow access to the fuel tanks, and with sensors and controllers to aid in the safe transfer of propellant. Studies to understand the needs and risks of autonomous refueling should be a funding priority in the near term for NASA, with continued investment into developing this technology, or adapting satellite servicing technology, if it is found to be the appropriate path forward.

3.6.6.2 Autonomous Algorithms for Propulsion Functions

Various algorithms are required to analyze the sensor data, determine system health, follow procedures or create action plans, and account for automated processes like refueling. This section will describe the developments needed for the propulsion subsystem.

3.6.6.2.1 System Health Management

System health management is needed across many subsystems, and is covered in Section 3.13.2. In particular for propulsion, the identification and localization of a leak is required using sensor data and reduced situational awareness. While the emergency response subsystem handles leaks internal to the spacecraft for the health of the crew and other internal components, the propulsion system requires the leak detection in valves and pipes that are largely external components. Because of the extra complexity required to do that, fault detection algorithms that involve small perturbations of the system (i.e., opening or closing valves to collect data to determine the precise location of a leak) could provide the appropriate ability to determine system health without requiring an extensive sensor network. The increased complexity of the active diagnosis algorithm would reduce the overall cost of autonomy in this subsystem. However, investment is needed to push these low TRL technologies to an appropriate robustness.

3.6.6.2.2 Task Planning and Procedure Execution

While many systems will require various task planning technologies (covered in Section 3.13.3), propulsion will also need the ability to follow procedures. Pre-defined jet selections will be deployed to the system for nominal and contingency scenarios, and these should be followed systemically. However, for maintenance, these tables may need to be circumvented in order to accomplish the task. A provable way to ensure that the maintenance procedures are followed without sacrificing safety or autonomous response to faults is required. NASA is already investing in procedure execution technologies, and the TRL should be advanced as needed.

3.6.6.2.3 Robotic Refueling

The potential need to refuel the spacecraft for the return trip requires some way to accomplish this process. Several other technologies that are already discussed as part of this document will be required for this task. Examples include AR&D and robotic manipulation during the attachment and detachment of refueling couplings. Developments needed for these functions are outlined in Section 3.7.6.

Certain process technologies required for the refueling task are not covered elsewhere, however, such as the method used to control the transfer of propellant. The pumps, valves, thermal and pressure constraints, and other mechanisms and considerations required for this process must be carefully engineered as part of the design of the overall system. The algorithms for this process must also be considered when developing the interfaces in order to ensure that the integrated solution is designed for the task. NASA's investment into these technologies is crucial for mission success.

3.6.7 Criticality and Redundancy

Propulsion systems are for the most part Criticality 1.   These Criticality 1 items consist of tanks, lines, active components, engines, and instrumentation. There may be instrumentation that is Criticality 2 or 3, but the majority of instrumentation is also Criticality 1.

In order to meet a reasonable reliability number of 0.99 or better, a propulsion system requires redundancy based on historical experience with propulsion system active components and engines. Given the long duration and difficult environments of exploration missions, active component and engine redundancy becomes more necessary to meet reasonable reliability numbers. The expected fault tolerance is single fault tolerant (1R1) for active components and engines. Tanks and lines (passive) are typically not redundant, but may be isolatable using valves. Instrumentation may be Criticality 1R1, given that these typically are not as reliable as other propulsion components. Diverse redundancy in a measurement or detection capability is preferred.

3.7 Robotics

Robotic systems will be necessary for supporting dormant spacecraft for tasks that require mobility, manipulation, or inspection capabilities in the absence of crew. These tasks may include preventative or corrective maintenance, logistics management, or fault recovery. Robotic assets may be used for both IVAs and EVAs, and more than a single form factor may be required to accomplish the needed tasks. The three robotic capabilities are further defined as follows:

1. **Mobility** involves moving supplies or robotic assets from storage to worksites and back again. Mobility could involve climbing, free flying, or moving along a structured transit system.

2. **Manipulation** involves capturing, grasping, and manipulating modules and payloads, unpacking, stowing, and repacking logistic modules, mating and berthing in-space modules, connecting intricate components (such as electrical and fluid connectors), opening panels and covers, and conducting maintenance functions. Manipulation tasks may be gross or fine, involving a range of dexterity. Manipulation tasks may involve tool use.

3. **Observation** involves compiling sensor data into a unified model of the state of the environment and the robot in that environment. Sensing and perception are essential to observation, but the synthesis of the information sensed and perceived is an important aspect of the observation capability. Observation may involve sparse sensor arrays working in concert with a mobile sensor package that has increased resolution or accuracy.

3.7.1 Nominal Operations

Many scenarios for robotics operations are likely for a dormant spacecraft. First, inspection of the spacecraft will be necessary in the absence of crew members. Inspection, both visual and other modalities, will be required for any unexpected, off-nominal events. Events requiring inspection certainly require observation, but may also require mobility and manipulation; for example, if an event requires a measurement to be taken behind a panel or under a blanket, a robotic manipulator may be necessary to enable the inspection. Similarly, if an inspection needed different or better sensors than the array that is currently providing data, more capable sensors may need to be mobilized to the appropriate locations. Depending on the design on the spacecraft, inspection may be needed both inside and outside the spacecraft.

Another scenario is maintenance. Robotic mobility may be required to retrieve the tools and spares needed to provide maintenance. Manipulation is necessary to use the tools, to configure the system, or to change out old or failed components. Observation may be needed to identify the amount or type of maintenance that may be needed. As an example, a filter may need to be inspected before a decision is made to replace or clean it. Observation is required to collect and synthesize the data required to make the decision, and then manipulation would be required to do the cleaning or replacement. Mobility may be required to either gather the cleaning apparatus or retrieve a spare filter. Depending on the design of the spacecraft, similar maintenance tasks may also be required outside the vehicle. Logistics management and emergency procedure execution are two other scenarios that robotic support may be required in nominal dormancy operations.

The concept of operation for a robotic system would start with procedures and information about the task being supplied to the robotic assets required. The robotic assets would then execute the procedures, asking for support, clarification, or confirmation when required. Depending on the complexity and criticality of the task being commanded, the procedures may come from vehicle systems manager, from ground control, or even from crew members on the Martian surface. Likewise, the support, clarification, or confirmation may come from any of those three places. Control modalities such as telepresence, supervisory control, or autonomous operations may be required for the robotic system, and may change depending on the task at hand.

3.7.2 Transition Operations

During crewed times, the uses for external robotics likely remains the same as for dormant times. However, internal robotic assets may not have as many or any functions required during crewed periods. Crew members will be more capable at inspecting or providing maintenance support. Limited logistics or emergency procedure support may be required, but this will certainly depend on the ability of the robotics assets to complete tasks in a timely and unobtrusive manner. As such, a possible set of transition activities that have to do with the stowing, unstowing, configuration, and checkout of internal robotic assets will be discussed.

3.7.2.1 Preparing for Dormancy

External robotics will likely be used in a similar way as during crewed operations while the spacecraft is dormant, except in the case of tele-operation control modes. Any tele-operation control station on board the spacecraft will either need to be shut down and stowed or packed up for the trip to the surface, depending on the plans for using the station during surface operations.

Internal robotics will need to be unstowed, if they were not used during crewed periods. The associated docking station for charging and other control interfaces, if needed, will also need to be deployed. The robot(s) will need to go through a full checkout, and any maintenance needed to support their operation will need to be performed. If the robots were used during the crewed stage of the flight, then any internal commanding stations will need to be powered off or configured correctly for the dormant period.

3.7.2.2 Transition out of Dormancy

Transition back into crewed operations will require the reverse of the steps listed above. Any tele-operation control station for external robotics will need to be unstowed and configured, likewise for control stations for internal robots used during the crewed period. Internal robots that are not needed for crewed periods will need to be powered down and stowed, along with any of their peripherals.

3.7.3 Contingency Operations

The operations that will be required in case of a fault or failure of the robotic systems will depend on the criticality of the robotic asset. For example, robots that fulfill critical roles in maintaining or operating the dormant spacecraft will have a different set of contingency operations from those robots that are fulfilling supporting roles.

3.7.3.1 Contingency Operations for Critical Robotic Systems

Internal or external robots that are fulfilling critical roles in supporting the maintenance and operation of spacecraft subsystems must support fail-operational modes. Robot failures are hardware dependent, but generally, failures in actuation, sensing, or computing are typical. There are several ways to mitigate failures to create a more robust system. First, there could be redundant lines of sensing. Sensors are typically fairly small, but can take a significant processing and/or power load. Smart distributions of sensors or fail-operational strategies can be chosen to minimize this effect.

Second, there needs to be redundancy in actuation. While it is likely not cost effective in mass, complexity, power, or processing to have a redundant actuator at every joint, functional redundancy could be designed into a system by having more degrees of freedom than what is needed or by having chain redundancy. For the first example, if it is possible to lock the broken joint and determine the joint angle at failure, the manipulator can still accomplish its task even with the broken actuator. For the second example, a two-armed robot has redundancy by virtue of its two arms and end effectors (as long as the tasks can be completed with one arm).

Finally, there are computation choices that can be made in order to allow robots to operate robustly. Robotic control algorithms are currently power and processor-hungry, making current radiation-hardened processors infeasible for the types of tasks that are currently planned. However, future iterations of these processors may be sufficient for the task in the future, and using these processors would reduce the failures due to cross-checking calculations and measurements. Assuming that radiation hardened processors will be feasible in the future, redundant processors could be used for separate tasks and algorithms (i.e., machine vision or path planning) and ready to fail over into the controlling role when necessary. In this case, less essential functionalities will need to be shed in order to allow the robot the proper functionality to recover the error through selective power cycling or other recovery function.

Failures in robotic systems could happen while in use or upon initialization. Failures in initialization will be handled through planning for sufficient lead time before a critical task must be completed. During the initialization checkout phase, any suspect or failed hardware will be diagnosed and verified. If a repair is possible for the failed hardware, that repair will be attempted. Possible repairs could include restarting or replacing the component. Common hardware modules that are susceptible to failure will require easy access for robotic repair. Robotic repair could be conducted by the robot itself (using a redundant manipulator, perhaps) or a secondary, supporting robot. Types of components that may require repair or replacement include motor controllers, end effectors, or vision sensors. Components that are not designed for in situ repair or replacement will require a redundant mitigation plan.

Faults and failures that occur while the robot is in use will be handled according to the type of task it is performing. If the task is not time-critical (like preventative maintenance), both the repair and fail operational options are available. However, if the task is time-critical, fail-operational contingencies must be used, with the repair option held for after the task is completed.

3.7.3.2 Contingency Operations for Supporting Robotic Systems

The robotic assets that do not fill critical roles, but instead provide support for contingencies that may be two or three failures deep will have fewer restrictions on their possible failure responses. In general, supporting robotic systems may follow everything that is described above for the critical robotic systems with the exception that, the fail-operational mode is never required to be used. Operations with reduced functionality are possible, but extrinsic redundancy is unlikely for these systems.

3.7.4 Preventative Maintenance and Logistics Support

Robotic assets will require periodic checks of their functionality in order to ensure the systems are available for use when required. These checks will involve testing out their functionalities, and if anomalies are found, corrective maintenance will be required. However, the robots should ideally be designed such that no preventative maintenance will be required, or that any preventative maintenance can be conducted via the robotic assets that exist.

Operational considerations for robotic asset needs are required, however, for things such as recharging or refueling. Ideally, the robotic assets will be deployed with a docking station that will provide the charging or fueling needs. Robotics assets should be designed such that their power or propulsion needs are available in the spacecraft’s supply of renewable resources so that logistics support for the operation of the robots is not required, however, for certain assets that may not be practical. In that case, fuel may be required as a logistics item.

Spare parts for the robotic assets may be required depending on the criticality of the robot and the likelihood of failure. One design option for spare parts is to 3D print the necessary parts from stock on board the spacecraft. The robotic asset must be designed to accommodate this type of spare, and it is possible that print stock could be generated from waste.

3.7.5 Autonomous Functions Needed

3.7.5.1 Nominal Operations

Nominal operations are dictated by the MTBC and MTBT for the tasks. For each task, the MTBC and the MTBT are currently on the order of minutes. Robots require nearly constant supervision and commanding during their active periods. For each task, the desired commanding frequency is on the order of a day or more, and telemetry is desired to be checked daily. Technology gaps identified may cross-over between tasks, but are just described once.

3.7.5.1.1 Mobility

Mobility commands to emplace assets for logistics, maintenance, or inspection tasks are expected for mobility systems. Given the desired MTBC of days, commands to mobile robots would have to be high level, such as “Deliver object A in location B to location C by time D.” The ability to autonomously execute this simple command is the driver of the following technology needs.

3.7.5.1.2 Path Planning

Mobile robots must be able to plan a path through the environment. This path must be feasible with respect to the space and with respect to the robot's abilities. This ability to find a motion plan with respect to constraints is an active area of research. Current motion planners are limited in degrees of freedom and number of constraints they can solve for. They also tend to be computationally expensive, time intensive, and often require significant human interaction to use.

3.7.5.1.3 Localization

Mobile robots require a map of the environment, either given, self-generated, or a combination of both, in order to understand where it is in the environment. Localization within this map is required to specify where tasks must be completed; visual and other sensory inputs are needed to match the actual robot's location with the desired location. Currently, map creation and localization within that map is fairly well understood in many types of environments. However, GPS-restricted indoor environments are still challenging, particularly if the environment has elements that can change.

3.7.5.1.4 Obstacle Avoidance

Mobile robots must be capable of identifying and avoiding obstacles in their environment. The ability to plan or replan around obstacles, once they have been sensed, is a topic of research. Obstacle avoidance requires vision processing, sensor fusion, localization, and constraint-based motion planning. The current state-of-the-art in autonomous cars is enabled by powerful sensors and processing techniques, both of which must be adapted to a space environment, and is aided by relatively simple two degree-of-freedom motion planning algorithms in a carefully mapped and GPS-enabled environment.

3.7.5.1.5 Manipulation

Manipulation commands to retrieve, replace, or repair objects are expected for this reference mission. Given that the desired MTBC is days, commands to manipulators will have to be at the task level. The ability to autonomously execute manipulation tasks is the driver for the following technology needs.

3.7.5.1.6 Tool Usage

The ability to grasp and use tools is an important part of manipulation. In very simple cases, end effector switching can enable direct tool usage in a pre-programmed manner. For manipulators that are meant to be able to service a great many interfaces using many types of tools, it may be more efficient to provide one or two types of end effectors that can interface with many types of tools. For these "generalist" manipulators, it is then required that they can identify tools, understand how the tools are used, plan the grasp to acquire and then use the tools, and then provide the appropriate control system parameters to most effectively use the tool. For example, soldering connections to repair a broken electrical board and using a torque wrench are very different types of tool manipulation. A general manipulator could be effective at using both of these tools given the abilities described above.

The current state-of-the-art in the switching manipulator case is the satellite servicing demonstrations, such as Raven, shown in Figure 6, that have been conducted on the ISS. These robots are specially designed to service the interfaces found on the outside of the ISS or of satellites. The mass required for this manipulator is greater since all of the tools must be carried with the robot; however, for interfaces where maintenance will not be typically shared with humans (i.e., EVA), the reduced complexity of these robots may outweigh the cost in mass.

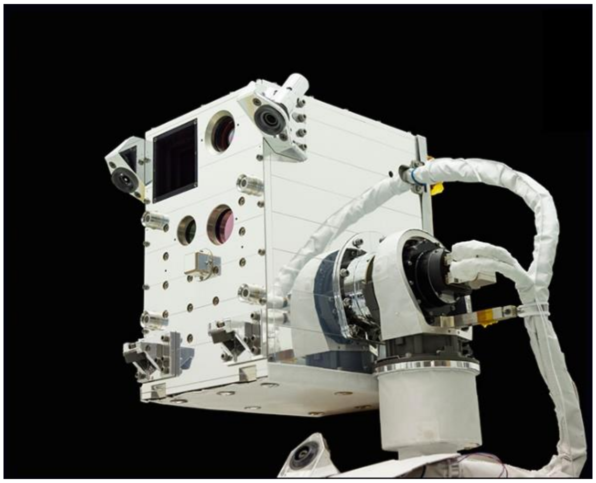


Figure 6: RAVEN

The current state-of-the-art in generalist manipulators is the Robonaut system with the affordance template framework. The dexterous hand is able to grasp many types of tools and interfaces, and able to use and actuate a smaller subset of those tools and interfaces. The affordance template framework allows operators to quickly inform the robot what sort of grasps and path plans will be required to both acquire and use the tools. The control system of the robot is such that several types of tools and interfaces can be used, including precision tools and interfaces that require compliance. However, more effort is required to create a robotic manipulator capable of both compliance and impulse, precision and dynamic range. Furthermore, many advances in the affordance template system are required to make this promising technology more useful and autonomous, including incorporating learning techniques to autonomously expand the corpus of tools and interfaces the manipulator would be able to access.

3.7.5.1.7 Object Recognition and Localization

Manipulators are concerned with objects, tools, or interfaces. In all cases, the proper object must be found in the environment and then localized within the robot's frame of reference. The recognition of objects is a task that has a range of difficulty. In highly structured environments, prior knowledge through models or some sort of pre-defined marker on the object can aid in both object recognition and localization. In unstructured environments, it is not always possible to have a marker on every tool or interface. In that case, other techniques are required to detect, identify, and localize objects.

Current state-of-the-art in machine learning for vision processing can be found in most light industrial manipulators, where a camera in the end effector can be directed to "learn" the type of object that needs to be manipulated. Unfortunately, this technique is not robust, as seeing objects from various angles can cause errors in identification or localization, and it is prohibitive to transfer this learning between instances of the robot, due to the reliance on calibration that is implicit in the learned model.

Model-based vision processing is a technique that has promise, though much work is needed to ensure a common algorithm can be used for various types of models. Learning will certainly be a part of future efforts to identify and localize tools and interfaces, and the process of sharing what is learned amongst instances of the manipulator is an area of future work.

3.7.5.1.8 Task Planning

Manipulators will be commanded to complete a task. This may involve following a procedure that has several steps. However, most procedures written for humans assume a large amount of "common sense" knowledge. For example, the procedure step to "vacuum the filter" in the crew's HEPA filter cleaning procedure would require a detailed task breakdown if a robotic manipulator was required to complete that step. The robot would need to be encoded with the way to hold the vacuum hose, the path to take to ensure the coverage of the filter, and other information such as how close to the filter the vacuum hose should get or how hard it could push on the filter. The breakdown of this information into specific steps that respect the constraints between the steps (such as temporal, spatial, or resource constraints) is task planning.

The state-of-the-art in robotic task planning requires low numbers of states and constraints to be computationally feasible. Current techniques rely upon knowledge bases and have limited success with learning techniques. Research into both understanding how activities could autonomously be broken down into tasks as well as research into the techniques to create the constraint-based task plans are needed.

3.7.5.1.9 Sensing and Inspection

Commands to inspect an area or piece of equipment will involve directing a sensor platform conduct a survey of the area or object of interest. As the inspection will likely include observing something that could be unexpected, and the MTBC is days, there are several autonomous tasks, beyond mobility or possible manipulation needed to reach the area or object of interest, that will be needed.

3.7.5.1.10 Intent Recognition

Robots do what they are commanded to do- but often, in human-robot interactions, humans find that their model of the robot's model is incorrect or incomplete, resulting in unexpected behavior. Inspection robots, which are likely to be sent into unmodeled and unknown situations, must have the ability to understand the intent for which they have been commanded in order to provide useful data back to the human operators. For example, if an inspection robot has been sent into a module to understand the environment when a small leak has been suspected, the robot should also know enough to report any fires it might find in the area.

The ability to command a robot in a way that the human's intent is realized is an area of research, and in general falls within the area of human-robot interaction. Various ways to approach this include creating interfaces that communicate the model of the robot effectively to the operator, or to give the robot the artificial intelligence required to understand emotion and intent. Both are active areas of research that must be continued to realize the abilities required of autonomous robotics for this reference mission.

3.7.5.1.11 Situational Awareness

Situational awareness is a requirement for any robotic system in an unstructured environment. This gives the robot the awareness of what is going on around it. While this is important for inspection robots so that they are able to determine the focus of their mission, it is also very important for the human operator to judge the success of the task a robot has been given.

Situational awareness requires advances in sensor technology as well as data synthesis and communication techniques.

3.7.5.1.12 Smart Telemetry Management

The ability of the inspection robot to understand its task well enough to send down only the information pertinent to the human operator is an important but difficult problem. Given the time delay and limited bandwidth, this may be critical to getting good data. While this is related to intent and situational awareness, this specifically addresses the technology needed to tie together the intent and data collected and make decisions based on that.

Research is in the very early stages for cognitive information synthesis (CIS) for robotics systems.

3.7.5.1.13 Note on Time to Criticality

Using robotic assets as mitigation for contingencies that have relatively short time to criticality may be desired, and as such, development is required to enable fast and robust robotic responses in the absence of continuous human supervision. This simply directs the level of robustness and autonomy of the robotic technology development needs that have been outlined above and does not add any extra technologies to nominal operations of these robotic assets.

3.7.5.2 Transition Operations

Transition operations for robotic assets involve the same tasks and autonomous technology needs as would be required during the nominal operations time. The act of shutting down or stowing would require the same functions as completing other mobility, manipulation, or inspection tasks.

3.7.5.3 Contingency Operations

For contingency operations to correct any robotic system faults or failures, the same autonomous technology needs are required for all of the tasks. The MTBI for each of the tasks are on the order of hours or days, and the desired MTBI is on the order of weeks to months for the reference mission. This drives several autonomous technologies and advances in systems design in order to achieve this.

3.7.5.3.1 System Health Monitoring

The ability of a system to identify and diagnose fault conditions and failures has been a topic of much research. Robots are complex cyber-physical systems that would benefit greatly from this technology, however, most robots are not equipped with the means to do health monitoring beyond annunciations of simple faults.

In particular, it is important that these robots be able to make decisions about their health in the face of these faults. For example, system health monitoring should be able to determine whether a fault was spurious or serious, and give the control system recommendations on next steps based on that.

In general, system health monitoring has been a passive activity, allowing slightly more insight into systems than the pre-programmed errors codes could provide. Many systems are model-based, and fault trees and models are required to provide accurate representations of system state. Very few robotic systems can give recommendations based on data or models, and more work is required to enable these systems to autonomously learn new operational norms given changing environments or operational dead bands.

3.7.5.3.2 Fail-operational Safing Response

The ability of a redundant robotic system to fail to a state of reduced operational response is important for an autonomous robotic system. The fail-operational state may be required for returning to power or for self-repair. The ability of a control system to accept data and recommendations from the system health management and to reconfigure the system autonomously in order to complete a small set of high priority functions is an important area of research.

3.7.5.3.3 Functional Redundancy

Mobile robots (including mobile manipulators) must strive to maintain redundancy in a small system package and mass. Therefore, functional redundancy is a technique that allows for the robot's design to have built-in ability to provide similar functionality upon failure. Some examples of this are to provide two robotic arms on a platform, so if one arm fails, the other is still capable of completing tasks or repairs, or adding in degrees of freedom to the system in the event that a joint fails.

Exploration of these sort of design paradigms has happened at a high level. Further research into the possibilities of functional redundancy and its effect on fail-operational safing responses is needed to feasibly deploy robust robotics solutions.

3.7.5.3.4 Modular Robotics and Self-Repair

If functional redundancy and the capabilities of fail-operational control systems are not adequate to ensure the required robustness of the robotic systems, exploration into modular robotics and self-repair may be required. Just as the robot may be expected to replace components that have failed on other subsystems, the robot could also replace failed components on itself.

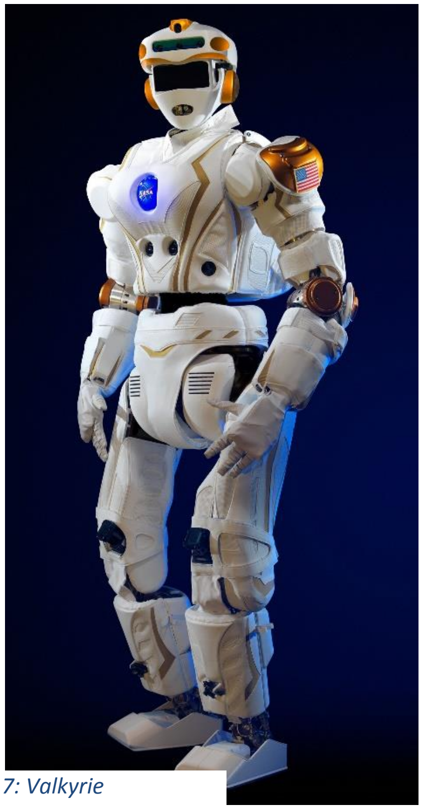


Figure 7: Valkyrie

The state-of-the-art is Valkyrie (shown in Figure 7), whose limbs require only a minimal interface to remove. However, more work is required before robots could successfully repair itself.

3.7.5.4 Preventative Maintenance and Logistics

Robots tend to be fairly complex cyber-physical systems that have many components and moving parts. As such, maintaining robots currently proves to be a significant task. Some of the technology advances needed for creating self-maintaining systems are listed in the Contingency Operations section, but having a capability to assess declining health of the system through prognostics is another technology that would ensure robust and reliable robotic assets for this reference mission.

3.7.5.4.1 Prognostics

Prognostics is a step beyond system health management in that data trends identify potential problems in advance. Prognostic tools exist for some systems (e.g., batteries), but have not been deployed in any meaningful way on robots as of yet.

3.7.6 Development Plan

The approach to developing and integrating autonomous robotics into a dormant spacecraft is two-fold. First, the habitat system should be designed for both robots and humans. Small changes in the spacecraft can ensure a less complex, more robust solution for the robotic assets that will be available to handle contingencies, maintenance, and logistics. Second, robotic technology must be advanced from current state-of-the-art. This development can strongly leverage various industrial and university efforts that are ongoing, but particular attention will be paid to the needed advancements that are largely limited to space applications.

3.7.6.1 Systems Engineering Design for Robotics

The opportunity to design both the spacecraft and the robotic caretakers together represents a powerful advantage in systems design to make the entire system more tightly integrated. Research has shown that robots can interact in a human's world- but they are vastly more successful and robust when their environment has been adapted to them. There are simple ways to ensure that this is the case, some of which will be listed below.

3.7.6.1.1 Mobility

Section 3.7.5 described three technologies that have impact on robotic mobility. Each of these technologies can benefit from smart system design. Path planning has been shown to be faster and more effective when past experiences can be used to seed the planner. Spacecraft can be designed to provide similar experiences for robots; for example, if a robot uses footholds to traverse, these can be placed at regular intervals. Or, for free-flying robots, no-fly zones can be implemented consistently throughout the spacecraft. While providing a rail system throughout the spacecraft may be prohibitive from a mass perspective, difficult to access areas, such as traversing a hatch, could have some special accommodations to aid robotic access as necessary.

Localization of the robot within the spacecraft can be simplified by using a map and by creating markers within the spacecraft. Given that different sections of the spacecraft are labeled in an easily observable way, the localization problem becomes much simpler. These labels could simply be AR codes that are placed in known, modeled locations.

3.7.6.1.2 Manipulation

As the robotic assets will be taking the place of human crew members during the uninhabited phase, it is important that these assets can do the tasks that are needed. In general, if there is a tool or an interface on board that may be needed to complete maintenance or emergency repairs, then the robotic asset must be able to use these tools and interfaces as well. While technology could be developed to create dexterous human-like grasping, a better systems engineering approach would be to design the tools and interfaces so that they are easy for both humans and robots to use. One example of this would be to create tools that do not require complex in-hand manipulation- while humans are quite adept at this skill, robots still have only primitive abilities to do so. Likewise, detection of tools and interfaces would be aided by appropriate distinguishing features, such as high-accuracy RFID tags or AR codes.

For task planning, designing the system so that the structure of most maintenance procedures is the same would reduce the computational complexity needed to accomplish an activity. This could be accomplished by providing similar interfaces for subsystem such that the large parts of the overall maintenance procedure could be reused from task to task.

3.7.6.1.3 Inspection

The situational awareness of a robot is a large contributor to task success. Given that the spacecraft is a controlled area, many system design considerations could be leveraged to create a beneficial environment for the robot. First, the state information generated by the spacecraft must be available to the robot. An appropriate communications and data sharing architecture can significantly increase the effectiveness of this. Second, the sensors on board the spacecraft should be used to provide further information and data verification to the robot. By providing the robot a "bird's eye" view, much more situational awareness can be accomplished by the entire system.

3.7.6.2 Robotics Technology Advancement

There is a long list of technologies needed for autonomous robotics in this reference mission. This section is ordered by expected investment required by NASA into the specific technologies. Table 14 summarizes the investment recommendations and priorities described in this section.

3.7.6.2.1 Intent Recognition and Cognitive Information Synthesis

One could argue that intent recognition and CIS are two sides of the same coin. In both cases, the developments needed for these technologies require the system to acquire data (or commands) and reason about that data using a model. For the intent recognition, the robot needs to have a model of the entity issuing the commands to be able to understand what priorities are. For the CIS, the robot needs to have a model of nominal to be able to understand what off-nominal looks like.

Technology developments for both of these tasks should be based around using models, whether learned or explicit or some combination of both, to understand a set of data autonomously. This work is low TRL, but is being conducted in the field of robotics and artificial intelligence. The initial technology developments in these fields will be best conducted by industry and academia, however, since the models that will be used for the application that this reference mission dictates are specific to this situation, NASA should be prepared to fund the research required to increase the TRL and infuse these technologies into a form sufficient for spaceflight. For example, some solutions require cloud computing services- the model ontology for this reference mission will need to be stored on board the spacecraft. Therefore, novel model abstraction and storage techniques may need to be developed.

3.7.6.2.2 Robust, Fail-Operational Robotic Design

Robust and reliable robotic design is an aim of the robotics industry today, however, they are also interested in mass production and cost. The robots that will be developed for the specialized application of doing mobile manipulation or inspection on board a spacecraft in orbit will need to be robust, reliable, and light. The conditions that these robots will work in is sufficiently different from ground applications of robots that industry and academia are unlikely to find a complete solution to the technological problems that will be faced by these robots. As such, NASA needs to fund the development of robots for robust, redundant, fail-operational design.

NASA needs to fund the development of hardware testbeds, new robotic systems, and software algorithms for fail-over redundancy. These testbeds will advance end effector robustness, reliability and fault tolerance; joint sensing and fail-over capabilities; radiation tolerance of sensors and electrical systems; and communication bus reliability and robustness to failures. The new robotic systems will integrate new sensors, electronics, joints, and end effectors into a single package so that further development of the reliability and robustness of the system can be tested. Furthermore, proper system design of the robots will allow NASA engineers to invent the capabilities needed for module self-repair and fail-over control systems. Several iterations of these robots will be required to have the technology available to deploy these robotic systems. Typical design cycles will take 2-3 years, with the first several months to a year being spent on the testbeds so that concurrent hardware and software development can happen, culminating in an integrated system. These integrated systems will provide for excellent technology development platforms for the various other technologies that are listed here.

Table 14: Investment Priorities for Robotic Technology Development

Technology	Priority	Notes
Robust, Fail Operational Robotic Design	Highest	Requires hardware iterations every 2-3 years. Requires space hardware, launch, and test.
Tool Usage, Object Recognition and Localization	High	Adaptation for space may be significant
Task Planning	Medium	Supplement current research and adapt for space

Intent Recognition and CIS	Medium	Adaptation for space will be significant; full functionality not required
Obstacle Avoidance and Situational Awareness	Low	Foundational work- development of sensors for space environments
Path Planning	Low	Foundational work- adaptation of current work for space is moderate

3.7.6.2.3 Task Planning

Likely developments in task planning for robots will require large amounts of data to be able to accommodate the significant amount of prior knowledge about the task or activities to be performed, or require significant computational ability to resolve the constraints. In either case, task planning for remote robots with access to more limited data and computational resources will be a challenge. Task planning is one of biggest gaps currently facing autonomous robotics. Though it is expected that commercial drivers for task planning will push technology in this field, it is unclear that the timeline will be suitable for current future exploration mission plans. Therefore, while NASA's investment in developing this technology should focus on reducing the overhead required, investments in robotic task planning in general would provide a driver for this important technology. It should be noted that while task planning is a technology that is used for autonomous systems outside of robotics alone (i.e., spacecraft control or crew scheduling), robotic task planning differs slightly in the need for "common sense" knowledge required for the physical tasks the robots do.

3.7.6.2.4 System Health Monitoring and Prognostics

Developments in system health monitoring and prognostics are needed primarily in the ability to accommodate complex systems and to adapt to changing conditions. While these technologies are in no way limited to just the robotic systems, the use of these technologies on complex robotic spacecraft and systems could be a stepping stone for these technologies on the larger integrated human spacecraft system. This technology is described in Section 3.13.2.

The primary driver for NASA investment in these technologies are the unknown environments and conditions into which the robots and spacecraft will be deployed. Without the ability to model the systems and the environment fully and likewise without the ability to train the system using relevant data prior to launch, technology advances will need to be made to allow the system health monitor to adjust in situ as "nominal" changes due to new knowledge and exploration.

3.7.6.2.5 Tool Usage, Object Recognition and Localization

The technology developments in this area involve using tools in unstructured environments. While algorithms have been developed to grasp objects in unstructured environments and to use tools in structured environments, the technology leap must happen in both manipulation and sensing for using tools in unstructured environments. Robotics research has focused recently on the "pick-and-place," but investment is required for the constraints of tool usage. Recent advances in disaster relief robotics has provided development in this area, and business cases for robotic caretaking and emergency management are starting to be realistic. It is reasonable to expect that the larger robotic research community will provide advances in this area in the near future. However, most of these applications are missing certain conditions that will require research, such as harsh lighting, radiation, and remote commanding over latent and low bandwidth connections. Investment in the infusion of community driven technologies into space applications is important to the future success of these missions.

3.7.6.2.6 Obstacle Avoidance and Situational Awareness

The advances in obstacle avoidance and situational awareness center around sensor technology and data analysis techniques. The obstacle avoidance problem is largely solved (or will be soon) in the advances needed for autonomous cars, however, the sensors used by these algorithms are generally unsuitable for space applications (i.e., large, radiation intolerant). Likewise, the algorithms themselves work through large amounts of data to determine representations of the environment. While work is ongoing to reduce the mass, power, and computational load requirements for these algorithms, the specific need for radiation tolerance is where NASA should focus investment.

3.7.6.2.7 Path Planning and Localization

Path planning and localization of complex robotic systems in GPS restricted environments is a challenge due to the computational requirements and algorithmic fragility of current solutions. While critical mass in the research community is available to solve these problems, NASA investment should attempt to focus the best solutions towards the demands of deployment on space robots. These challenges include limited computational ability, highly constrained environments, and limited access to human support.

3.7.7 Criticality and Redundancy

The robotics systems are necessary to ensure the health of the vehicle in the absence of crew members on board. Mobility, manipulation, and inspection functions are expected to be required in the case of the unexpected, as a line of mitigation against failures. For example, in the case that spares are on board the spacecraft for critical systems, the ability to change out those spares must be weighed against the added complexity and mass of a robotic system to do the repair. The criticality of the robotic systems in these functions depends on the criticality of the task they are commanded to complete. If the system must repair a breech in the spacecraft's structure and then recover the systems affected by the breech, then the robotic system would have level 1S or 1SR criticality overall.

Robotics may be used as a system for non-emergency tasks as well, such as logistics management and preventative maintenance. The failure of the system to accomplish these tasks has effects that are proportional to the criticality of the task. For example, if spares are stowed in unpressurized volume outside of the spacecraft, accessible only by robotic assets, the failure of these assets could be a level 1S or 2 criticality depending on the intended use of the spares.

Operationally, the time to criticality for the task the robot is completing dictates the operational criticality of the robotic system. For example, if a robotic asset is required to repair a breech, the time criticality of diagnosing the robotic system, if required, would follow from the importance of operational priorities dictated by the breech.

The robotic systems themselves carry criticality in the damage that could be inflicted by a failure. Current examples include Robonaut on the ISS. Robonaut has several hazards that have been judged to be critical or catastrophic, due to the damage that could be inflicted if a failure was to occur. As such, internal system design for fault tolerance was dictated by this criticality.

A failure of a robotic system locally affects the operation of the robot. The immediate effect is to the robot itself. The next effect has to do with the task or tasks that the robot is commanded to complete. The next effect could also be in response to the way the robot fails- in particular, if the robot fails in a hatch, for example, or fails by somehow applying excessive force to a sensitive interface, the next effect would be due to these things. The end effect depends on the tasks that are delayed or not completed due to the robot failure, or in response to the effects of how the robot failed. These effects must be carefully considered when designing the robotic systems.

For some robotic systems and tasks, the criticality increases when humans are not on board. Some robotic assets are expected to provide functions in the absence of crew- if crew were on board and available, it is feasible that the tasks could be carried out by them. For example, if an avionics box fails and needs to be replaced, a robotic asset would be required to do that task in the absence of crew. If crew was present, the maintenance could be conducted by them. As such, the criticality of the robotic assets is reduced in this phase of flight. However, if the robotic asset is performing a function or a task that cannot be replicated by human crew members, the functional criticality remains the same, though the operational criticality may be relaxed without crew present.

Robotic systems can approach the need for fault tolerance in several ways. Within the robotic system itself, robustness to failures can be accomplished through smart sensor selection, robust communication mechanisms, and reliable actuator design. Robustness in manipulators is demonstrated by the industrial manipulators that exist in factories worldwide. However, adding in requirements for safety or fail-operational control systems increases the complexity of the robotic system. As such, redundancy of sensing and computing is required. Redundancy of actuation is generally achieved through functional redundancy, i.e., extra degrees of freedom or dual manipulators. In these cases, though, design decisions must be made to ensure controllability in the case of actuation failure.

There is often a trade-off between robustness and complexity in system design, and robots are no exception. One strategy would be to have several simple robots to ensure that criticality and redundancy requirements are satisfied, however mass considerations often prohibit this strategy. As such, careful consideration and design of sensing, actuation, control and safety systems are necessary to ensure robots with the proper robustness, fault tolerance, and redundancy for the tasks required of them are chosen.

3.8 Software

Flight software that permeates and connects spacecraft subsystems, telemetry, and commands will be treated in this section.

3.8.1 Nominal Operations

This section summarizes software capabilities during nominal operation of a dormant spacecraft in-transit to a specific destination or on-station at the planned destination. While the spacecraft is in a dormant state, the Software System (or software) will be tasked with the following objectives.

The first objective is subsystem monitoring. In dormancy the spacecraft will require monitoring of all active subsystems to ensure spacecraft health and status is reported to ground control on a periodic basis. Subsystem monitoring will continue until the spacecraft is prepared and activated for the next mission phase.

The second objective is command handling and processing. In the event that software capabilities and/or data need to be updated to address potential performance or operational needs, the software will process commands specific to the dormant phase of flight.

In support of these objectives, a minimalist approach is taken. This approach requires the software to provide processing for dormant operations, i.e. the operational subsystems will provide capabilities only needed to safely maintain the spacecraft. This includes software capabilities to detect faults and failures but not fault isolation and recovery.

The software will produce only data that is needed to accurately characterize and communicate spacecraft health and status. All data collected is processed to reduce the impact to the telemetry system in terms of bandwidth and performance. No in-flight servicing or in-flight maintenance will be required or initiated by the ground or other spacecraft. Any and all "housekeeping" functions will be performed by the software without operator assistance.

The scope of the software system for the dormant spacecraft includes all spacecraft system and subsystem data processing, with the exception of board-level or component-level logic, e.g. VHDL or firmware.

3.8.1.1 Subsystem Monitoring

The objective of subsystem monitoring is to collect and process data acquired from active subsystems in the dormant spacecraft. Processing would include performing limit checks on key data items, summarizing subsystem health and status at the spacecraft level, and storing acquired data and health and status summaries for later access. During nominal operation, the software would report status periodically to the Ground and potentially to other spacecraft. This data set's scope (size and content) would be gated by dependencies on power, as well as telemetry bandwidth and data rates. The reporting cadence could initially range from 1 hour to 24 hours, but would be configured based on dependencies defined below. In addition, the Software System would send data to vehicle systems manager (VSM) based on nominal operational requirements or flight rules.

The active capabilities of the following components are expected to provide limited health and status data. Most subsystems will also provide key performance data.

- Avionics Hardware
- Communication
- ECLSS
- GN&C
- Power
- Propulsion
- Robotics

- Spacecraft Emergency Responses
- Structures
- Software System
- Active Thermal Control

Subsystems where monitoring or processing is not required are as listed. In a dormant spacecraft under nominal conditions these subsystems have been powered off or are passive in nature.

- Life Sciences
- Passive Thermal Control

3.8.1.2 Command Handling and Processing

Although unexpected, updates to the spacecraft's data and software may be necessary based on ongoing ground analysis and test. While dormant the software system would have the ability to process such updates, the updates may be limited only to flight software and flight data changes affecting the software that manages and controls the spacecraft during dormant operation. That is, the command set is restricted to only those updates necessary to operate a dormant spacecraft. Software updates that are "low priority" or support full operational spacecraft capability might be deferred until the spacecraft is successfully transitioned out of dormancy. The following command types would be supported as part of nominal dormant operations

The first command type is flight data updates. Flight data identified by ground analysis that would change the way a dormant spacecraft handles off-nominal situations or operation. Two general types of updates would include a) FDIR data to improve the software's ability to detect, identify, and recover from faults, and b) improvements in operational data to optimize VSM performance.

The second is flight software updates. Executable flight software updates identified by ground analysis that corrects high-severity defects that could potentially affect the dormant spacecraft.

The final command types are orbital maintenance and course correction. While the spacecraft is on station or in transit, these process commands and forward parameters alter the spacecraft's orbit or course, respectively.

3.8.1.3 Dependencies

Nominal power levels during dormancy will be reduced with respect to crewed operations and are the amount of power available to support GNC, Propulsion, Communications, and Avionics processors. This will determine the amount of data that can be collected and the amount of processing performed. A spacecraft under battery power would have significantly reduced software capability as compared to a spacecraft using RTGs or solar arrays.

Operational ground rules need to be defined to determine if and when flight data or flight software can be updated. It is generally a good idea to have some door through which data and software can be updated. However, limiting updates operationally would be a good idea to simplify onboard processing. Dormancy requirements (i.e. what really does a spacecraft need to be dormant) for each subsystem (component) on the spacecraft determines what portion of the subsystem is active, which drives monitoring and data requirements.

3.8.2 Transition Operations

This section describes the nominal ramp-up and hibernation software activities during operation of a dormant spacecraft in orbit. During ramp-up or hibernation, a number of spacecraft subsystems will be engaged requiring coordination and monitoring over a period of time. Regardless of the level of complexity or resource demand, the process and results of ramp-up and hibernation are expected to be deterministic, i.e. expected behavior and predictability in the startup and shut down of all subsystems is known with no variation to the process.

In the ramp-up process, the Software will execute a set of subsystem procedures (scripts or similar technology) that, on a high level, will power-up the subsystem, gather data to assess progress and report, command (if necessary) to confirm readiness, gather data to assess final readiness state and report, and finally, transition to the operational level or state of readiness selected. Subsystems are expected to perform as much of their initialization and self-test as possible, reporting progress at predetermined intervals or at specific break points.

The transition to dormancy is analogous to the ramp-up process, but decidedly simpler. It includes commanding the subsystem to shut down or transition to a predetermined hibernation state, gathering data to assess progress and report, and powering down the subsystem. Subsystems are expected to perform as much of their shut-down or hibernation transition procedures as possible, reporting progress at predetermined intervals or specific break points.

3.8.2.1 Conceptual Operational Levels

At a higher level the VSM initiates all transitions, delegating execution of the transition procedure execution to Software. Software in this context applies to software at the subsystem level as well as the spacecraft level. As such, the concepts defined apply to the software at the spacecraft level. All subsystems provide results that can be summarized and used by the VSM to assess progress, including faults or off-nominal conditions. Off-nominal ramp-up or transition to dormancy conditions requiring the VSM to address the complexity associated with recovery of ramp-up or dormancy operations are addressed later.

Figure 8 describes conceptually the levels of nominal operation, which assume involvement from the various subsystems to accomplish objectives.

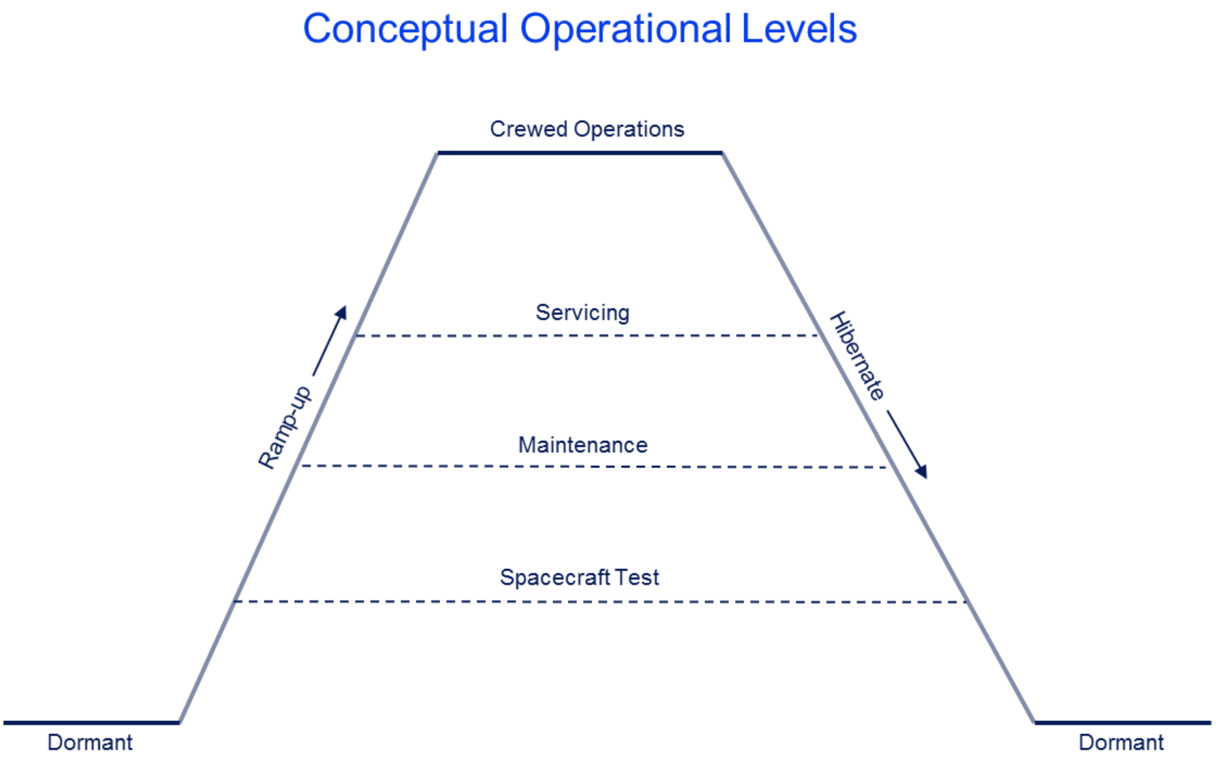


Figure 8: Conceptual Operational Levels

If the spacecraft is uncrewed for a longer period of time than expected or can be serviced by another autonomous or crewed spacecraft, these alternative nominal operational levels are likely.

The system test level periodically exercises the various spacecraft subsystems to assess readiness for full operation. System Tests can also be an integral part of the transition to a nominal operational level. Execution of a system test assumes some level of automated spacecraft assessment beyond subsystem BIT, e.g. determining whether power levels will be available to return the spacecraft to crewed operations.

Transition to maintenance operations allows the spacecraft to perform set of tasks keeping the spacecraft operating at the expected level. Tasks such as replacing filters, updating software, archiving data, cleaning the habitable environment to name a few may require activation of subsystems from their dormant states. Physical maintenance tasks (i.e. replacement of filters) could require activation of robotic capability.

Similar to maintenance, servicing involves performing a set of tasks to replace or restore consumables and involve activation of one or more subsystems. Propellant, air, water, coolant, etc. are examples of consumables that may require servicing. Physical servicing tasks could require activation robotic capability by the spacecraft.

The crewed operational level is the level at which all subsystems are fully activated to sustain a crew prior to arrival. The ramp-up and transition to dormancy procedures required to transition to and from this level would execute the full suite of procedures.

3.8.2.2 Ramp-up Procedures

The procedures involved to transition the spacecraft from dormancy to a full operational crewed state would involve an established procedure. Ramp-up procedures, would involve steps such as defining the operational level to achieve, preparing of the spacecraft by determining the current operational state (including confirmation that all subsystems are at their expected performance levels) and queuing up the procedures to achieve the next operational state, and executing the subsystem procedures to activate subsystems and complete initialization (keeping in mind that timing may be required to provide an orderly transition), complete BIT functions to ensure the subsystem is functioning properly, calibrate components for those subsystems that have sensitive instruments, and test and assess subsystem performance (i.e., is the subsystem ready to operate with other subsystems at the level specified).

Subsystem Procedures would typically follow a set sequence of events including controlling power switches to energize and activate subsystem components, accessing data sources to assesses progress and command paths to control the subsystem (if required), gathering fault/failure data to flag alternate paths for off-nominal processing (although nominal operations assume no faults will be encountered), monitoring the “critical path” required for ramp-up to ensure key subsystems are fully capable, and communicating with the VSM and logging status for later analysis and/or download to ground support. Completion of the procedures would involve deactivation of support subsystems, such as robotics that assisted in ramp-up operations during maintenance & servicing operations.

3.8.2.3 Transition to Dormancy Procedures

Conversely the procedures involved to transition the spacecraft to dormancy would involve a similar procedure albeit in reverse. These procedures, would be somewhat simpler and involve steps such as defining the hibernation level to achieve, preparing of the spacecraft by determining the current operational state (including confirmation that all subsystems are at their expected performance levels), and queuing up the procedures to achieve the next operational state, and executing the subsystem procedures to deactivate subsystems (allowing enough time for the subsystem or subsystem components to complete shut-down or arrive at a state of degraded operation) and provide a confirmation that the hibernation state has been achieved.

Execution of the subsystem procedures would typically follow a set sequence of events, including accessing data sources to assess progress and command paths to control the subsystem during hibernation (if required), gathering fault and failure data to determine alternate paths for off-nominal processing (although nominal transition operations assumes no faults will be detected), monitoring the “critical path” required for dormancy to ensure key subsystems remain active and healthy, communicating with the VSM and logging status for later analysis and/or download to ground control, and power switch control to de-energize subsystem components or the subsystem. Completion of the procedures would involve deactivation of supporting subsystems, such as robotics that assisted in maintenance and servicing operations.

3.8.2.4 Conclusion

The final nominal state of the spacecraft is the operational state selected. Arriving at this state, from a software perspective, could be accomplished by executing a set of instructions through a scripting engine. The scripts to perform nominal ramp-up to a specific operational level or transition to a defined state are predefined and deterministic.

3.8.3 Contingency Operations

This section covers software faults and recovery in a dormant spacecraft, providing a generally high-level but comprehensive perspective on software fault detection, isolation, and recovery. It is assumed the software, which includes data, has been rigorously tested such that any defects or bugs remaining are insignificant having no effect on the system or any subsystems. (This assumption relies on appropriate V&V techniques to be developed and applied to the flight software.) As such, “common cause” errors as a source of faults in redundant systems are ruled out.

3.8.3.1 General Software Fault Types

As a starting point, there are four types of software faults to consider. The first type is executable software errors caused by an alteration of the processing flow or logic. Although extremely rare in light of ECC capabilities, memory refresh techniques, etc., software specific error codes must be considered. Data faults caused by alteration or corruption of data values processed by executable software are more likely in a space environment.

The third is when software “goes silent” or is responding slowly. The use of software watchdogs can mitigate situations where the processor appears to have “locked-up” for one reason or another. Electronics induced faults are included since failures of this type will induce the vast majority of software fault recovery responses and actions. Fault responses range from power cycling a system via hardware watchdog (e.g. over current protection) to analysis of electronic hardware fault codes and data.

3.8.3.2 Notional Fault Levels

A general approach to organizing software faults is by recognizing levels of severity. Severity in this context is defined as the resultant effect(s) of the failed software and not the criticality of the system or subsystem impacted. Dormancy implies all operational subsystems are critical in nature and necessary for maintaining an uncrewed spacecraft. Refinement of the severity levels may be necessary depending on the complexity of the subsystems.

Minor faults are characterized by bad data or results produced by software that can be corrected on the next reporting cycle. There is no need to restart the software capability or capabilities due to the transient nature of these faults. Major faults describe software that continuously produces bad data or arrives at the wrong results that may eventually be detrimental to a subsystem. Although the software in question recognizes the issue, the software would require the next higher level of control to restart the application, e.g. a fault manager to unload from the software memory and reload. Critical faults are represented by software that is producing bad data and results thereby affecting spacecraft performance. Critical faults require the next higher level of control to restart the subsystem most likely by power cycling the hardware. If a redundant processing system exists, then control must fail over to the redundant system until the failed system can be recovered. Generally, the software would fault to a known safe state or mode, communicate with the next higher level of control along the way, and wait for the higher level of control (VSM or human) to intervene and recover. Catastrophic faults are distinguished by faults where the software system is lost completely with no apparent reversion to a safe state or mode. The software system is essentially lost and must be restored by a higher level of control (VSM or human).

3.8.3.3 Recovery Approach

For dormant spacecraft operation, all initial recovery approaches will be deterministic in nature, which requires set procedures that produce known and predictable results. Such procedures would incorporate decision trees to monitor and manage the recovery process starting from sets of known safe states. Safe states or modes are identified and built in such that the subsystem faults down to a level where recovery can take place.

The recovery approach is determined by the software's ability to gracefully degrade to a specific safe state. This safe state assumes the software system and spacecraft are able to tolerate the termination of non-vital functions, e.g. software that processes sensor data. Higher tolerance can be expected where redundant systems are available.

Generally, all levels of software will incorporate some aspect of recovery procedures, however the more sophisticated and complex recovery procedures will be executed by the VSM. As a ground rule, most subsystem software will pass recovery decisions to the VSM as their "higher level of control" to minimize complexity and optimize fault response and recovery actions.

Additional ground rules would limit a software application's ability to create recovery dilemmas. For example, a rule stating that no software system or software subsystem would be able to power cycle itself would eliminate run away processes, and a rule that says self-test capabilities would be used as part of the recovery process by the higher level of control to determine whether the software capability has indeed recovered would prohibit endless recovery cycles.

3.8.3.4 Conclusion

Analysis of software faults, identification of critical software applications and capabilities, and root cause and effects analysis will be required to define software recovery procedures and safing levels. The ability for the ground controllers to anticipate software faults that have yet to occur and update these procedures on a dormant spacecraft will require sophisticated simulation, fault analysis, and procedure development architectures.

3.8.4 Preventative Maintenance and Logistics Support

Preventive maintenance of software in a dormant spacecraft will primarily result from the need to anticipate and correct software defects, or to enhance the software improving spacecraft performance. Software preventive maintenance would not necessarily follow a periodic or set schedule. It also assumes preventive maintenance for mundane tasks such as compressing files, clearing disk space, freeing memory, restarting systems/applications, etc. is performed by the software on an ongoing basis. As such no intervention is required to maintain or provide service for any software application on a set schedule. Therefore, the goal of software preventive maintenance is to pre-empt faults, defects, or issues from occurring on an as needed basis.

3.8.4.1 Preventive Maintenance Concepts and Drivers

Since software preventive maintenance is primarily concerned with preventing faults, the basic concepts driving preventive maintenance for software include the following. First, the maintenance identified by the spacecraft is generally defined as faults or bugs trapped by the spacecraft flight software. General examples may include the wrong expected value(s) produced or erroneous action(s) performed. Bad data or a software failure resulting in the restart of an application or some aspect of the software system is an example of the extreme, however capturing information in the form of error indicators and fault data allows ground analysis to pinpoint and solve the issue's root cause.

Maintenance identified by ground analysis is generally defined as a fault or bug identified by ongoing ground analysis and testing on a high fidelity spacecraft simulator or like capability. Typically induced by flying sophisticated scenarios or analyzing specific use cases, this ability requires analysis methods and techniques that explore permutations or explore branches of the spacecraft's fault tree that may have been deemed low probability with moderate/high risk to the crew or mission. Software and related analysis technologies are required to help identify these issues, placing significant burden on the ground personnel to investigate and expose these problems before they have a chance to manifest themselves in flight.

Finally, data driven maintenance indicates learning that may occur during operation. The primary driver of changes to software behavior, including the ability to identify and recover from faults, would be encoded in or provided by data. Data is processed by executable software or "engines" to produce the required outcomes. Although this may be viewed as an architectural decision, the use of data preserves a level of determinism that will be required by the software to control the spacecraft while dormant in addition to reducing complexity and cost.

The maintenance identified by the spacecraft and by ground analysis, as well as the careful addition of new capabilities, features, or functions, are the major drivers for preventative maintenance.

3.8.4.2 Preventive Maintenance Updates

There are three general types of preventive maintenance updates for software. Most of the updates will involve the modification of data to affect software behavior, which includes alteration of script or procedures executed by the software. Further breakdown of these update types may be entirely possible depending on complexity.

First are updates to scripts and procedures supporting nominal and off-nominal operations. Those supporting nominal operations provide data updates to scripts or procedures required to maintain the state of the spacecraft or provide state changes during normal operations. Updates may implement new features and functions required to better support the mission or improve spacecraft performance.

Those supporting off-nominal operations provide data updates to scripts or procedures required to detect, identify and recover from faults during off-nominal operations. Although the updates are designed to improve fault identification and recovery processes for faults that have already occurred, the main emphasis is improvement and correction of recovery procedures for faults that have yet to occur in flight.

Updates to configuration data provide updates to data which sets the boundaries that software or hardware systems are required to operate within. Changes in this category improve or provide fine adjustments to mission performance as identified by ground analysis. Finally, updates to flight software provides updates to executable software that will correct defects and/or implements new features and functions. The primary objective is of course to correct defects and improve fault response, however software changes will be needed to better support the mission or improve performance of the dormant vehicle.

3.8.5 Autonomous Functions Needed

This section sets the context for autonomy gaps as identified for software. Since software is embedded in practically all systems and subsystems, this autonomy needs section will focus on the needs particular to the functions that software provides that are common across subsystems. A separate section will discuss needs for coordinating spacecraft-level autonomous functions, namely the VSM section. Operation and control of the spacecraft is foundationally dependent on the data collected from the various components and assets that make up each system and subsystem. Since data is the foundation for autonomous control, the next sections address concepts that provide a basis for software autonomy needs.

3.8.5.1 Dataflow Analysis: Foundation for Autonomy

Dealing with data across various subsystems, across various forms, and across various levels produces many interesting autonomy needs. As the spacecraft must be able to take on functions currently handled by a set of human ground controllers, the ability to collect, process, and give context to vast amounts of data requires dataflow analysis.   Data is needed for identifying and confirming success of mission procedures, for trend, rate, failure, and root cause analysis for maintaining spacecraft health, and for creating situational awareness for ground controllers.

For any complex software system, early efforts are devoted to developing a Software Concept of Operation. The Software ConOps must be developed in conjunction with the various data products required to operate the spacecraft from the control center. (This can be considered the mission operations software systems engineering task.) This data product development process will be ongoing throughout the software development cycle. Building the data products involves the methods and approaches as follows.

Dataflow Analysis is the identification and classification of data produced by each subsystem resulting in the building of a data dictionary or database of data. The objective is to capture every data element that is produced or used by the spacecraft software system. Analysis also identifies how the data that can be used for analysis during flight. This data must provide information communicating the state of spacecraft systems or subsystems or assets. Analysis is required for identifying the “right” data to telemeter. Optimizing the amount of data as well as the specific data to precisely communicate the current state of the spacecraft is extremely important in terms of performance and reliability. Methodical analysis is needed for identifying “alternative” or replacement data for key data elements. The objective is to develop networks of data that can substitute for these key data elements should they be missing, incomplete, or untrustworthy.

Subsystem data linkages establish and identify data dependencies and relationships within subsystems and between subsystems providing data network that identifies the direct and indirect dependencies between subsystems. A special case is Failure Modes Effects Analysis (FMEA) or perhaps Failure Modes, Effects and Criticality Analysis (FMECA). These products, provided by engineering for each subsystem, identify and establish direct and indirect fault data dependencies and relationships within subsystems and between subsystems to identify faults.

In the control center as well as onboard, software engines perform many different types of tasks. Some of these are data transformation (computations) to transform sensor data into more abstract information (engineering unit transformations, etc.), lower level device controllers (valves, pumps, fans, etc.), onboard computer networks, logging and data storage, and device or system FDIR.

Software configuration is used to develop data-driven “software engines” to every extent possible. Software Engines are designed to operate on configuration data, to the maximum extent, predict what may change and use configuration files which if changed, causes the software to behave in a different manner. This concept is the next level of table-driven software.

Much of the same dataflow analysis and resulting products will be used for managing the spacecraft with onboard applications. The difference is that there will be fewer computational resources onboard, and thus only a subset of the applications used in the control center can be deployed onboard. In cases where human flight controllers make decisions, new applications may be needed. On the other hand, the data need not be sent over a constrained communication link; all data can be used to make decisions as soon as it is produced.

3.8.5.1.1 Mission Timeline and Mission Procedures

Mission procedure execution carries out the sequence of tasks or operations that are to be performed to achieve the objectives identified in the mission timeline. In the context of this design reference mission, mission timeline will consist of system objectives needed for the operation and maintenance of the dormant spacecraft. Mission timelines for crewed spacecraft generally consist of both system and crew timelines, which can cross over when the crew is assigned to carry out a system procedure (such as planned or unplanned maintenance). Ground control generally handles the system timeline objectives that are possible without crew intervention. This section will focus on these system timeline objectives, which will be carried out by a combination of autonomy and ground control.

“Procedure” is a generic name for any set of tasks or activities that can be executed to achieve a stated objective. While most procedures are currently written for human execution, tools are being developed to create a procedural language that can be understood by humans or software. Further developments are required to create an autonomous procedure execution technology. However, regardless of the solution, data will be used and generated by the process of executing these procedures.

Data is required for the appropriate storage, annotation and identification of the mission procedures. Mission procedures are associated with specific failures, data signatures, or tasks that may arise. An automated way to associate vehicle state with procedures in a responsive and low cost way is needed. Data is also required during the execution of the procedure to provide the procedure execution engine feedback on the success or failure of the procedure. Procedure failure could be immediate or take place over a time horizon that extends forward from the procedure execution time. Data collection and analysis must be done in a manner that allows introspection into a procedure (i.e. procedure execution logs). Data could also be used to adjust procedures in situ, for example if sensor bias or error does not fit the model, and control inputs are commonly causing overshooting of the set point. Finally, data could be used to identify gaps in procedures as written. The collection and handling of data such that the above functions are possible is an autonomy gap that must be filled.

When multiple procedures are run simultaneously, they may need to be deconflicted. This is especially important when taking contingent branches or invoking off-nominal procedures. The procedures must be written in such a way that conflicts can be detected and resolved.

3.8.5.1.2 State Analysis

Data is required to understand the overall state of the spacecraft's systems. This includes data trend and rate analysis, failure detection, and root cause analysis. Nominal data trend and rate analysis is typically associated with spacecraft functions and behavior affected by the space environment, for example gravity fields or solar wind causing orbital decay. Off-nominal trend or rate conditions may arise as a hardware component degrades causing a quicker than expected effect on a subsystem. The data must be available for the software system to reason about it effectively. The system must also be able to recognize corrupt or incomplete data. Alternate data items should be planned, identified, and used in their place. Analysis of the data should provide these mappings beforehand. If alternate data does not exist, procedures will need to be capable of dealing with the situation, making decisions on the subset of valid data.

Data can be used for prognostic fault detection, which focuses on identifying data trends that will result in out-of-bounds or fault conditions. FDIR detects, identifies and recovers from faults. When device or system level FDIR is performed, the fault diagnoses and, more importantly, recovery actions must be logged and reported to the VSM (more on this topic in Section 3.8.5.2 below).

3.8.5.1.3 Data Logging and Downlink Management for Situational Awareness

Data must be autonomously organized and identified so that it can be interpreted by human consumers in an effective way. If the system experiences faults or failures, more information will be provided to the crew and ground control. This implies data throttling when required to provide pertinent information. Data provided is related to the subsystem experiencing faults, which implies a sophisticated form of variable downlist along with the ability to increase the data rate. This requires the ability to autonomously decide what data is required at what time to properly characterize the environment or state of the spacecraft. The system must also understand the telemetry data rates such that pertinent information can be supplied to the surface crew and ground controllers.

3.8.5.1.4 Software System Engineering's Emphasis on Data Analysis During and After ConOps Development

Software system engineering will not end with the spacecraft in flight. Operation of the spacecraft in the space environment will result in acquisition and identification of more knowledge, requiring ongoing data updates in terms of procedures, limits, operational constraints, etc. This requires a very robust and highly flexible simulation capability that can be used by operations and engineering to identify knowledge gaps and provide updates to the data inflight. A base assertion is that these data updates will be developed not only to change the mission profile, but more importantly to pre-empt faults, failures, and issues before they have a chance to manifest themselves on the dormant spacecraft.

If the areas above are addressed early in the spacecraft development cycle, the software autonomy gaps have a high probability of being resolved. The result is a spacecraft that can meet and exceed its autonomy drivers.

3.8.5.2 Locus of Control

Looking the spacecraft software system as a whole, there are differing levels of command authority and control. Starting with VSM at the top level, control authority can be delegated to subsystem software, which in turn can be delegated further to individual devices. One example of such hierarchical organization is software subsystems FDIR; a second is system-specific procedures. With this level of independence is established, there is recognition that lower levels of software cannot or should not attempt to resolve all faults or failures. In these cases, the lower levels of software should become advisory in nature providing data and recommendations to the next higher level for recovery while they transition to a safe state of operation. As noted above this organization imposes interface requirements on all components (the VSM and other software engines) to ensure the integrated system works well. Some design considerations are as follows.

First, each subsystem down to the component level has the ability for fault handling/resolution, however recovery is limited in scope within the subsystem, sub-subsystem, component, etc. When limits in fault handling have been reached, control of the fault is delegated up to the next higher level of authority; conceivably and ultimately arriving at the VSM or even ground controllers for final decision or resolution. Each subsystem provides recovery selection(s), if possible or practical, to the next higher level of authority who can break recovery dilemmas based on a broader understanding of the subsystem or spacecraft. All FDIR actions taken must be reported to next higher-level of control, including prognostic FDIR actions. The VSM will ultimately be aware of any fault no matter how minor, which provides a growing based of information for Prognostic FDIR.

3.8.5.3 Software and Software Configuration Updates

Updating software, which includes executable logic and a variety of configurable data products, is a critical activity due to the possibility of corruption during transmission or in-space. Although the software should be verified and validated on the ground before the software is ever transmitted to the spacecraft, bits could be flipped or lost in the process. As a result, the software will be loaded, executed, and monitored after updating to ensure correctness. Changing software configuration is less risky, but will still generally change software behavior; as such, these changes are subject to a similar level of rigor.

The following functions are required to ensure that the software update process is successful in this design reference mission scenario, software update verification to determine whether the update was transmitted without loss (e.g., use of checksums), software fault reporting and handling, automated software updates, and software partition(s) to test and prove the software load.

The technology challenges that result include determining whether the new version of software is faulty enough to require a rollback and maintaining version compatibility with software loads on other processors as various software packages are loaded and executed. Onboard evaluation of the update software loads will reduce the spacecraft's dependence on ground controllers for in-flight evaluation.

If this onboard evaluation indicates that the software is not functioning as well as the prior version, the VSM may decide to roll-back the software update. Although a remote possibility, all software must be designed to retain and reinstall the prior software version. This capability includes software fault analysis to determine whether the software is operating reliably and software data for ground and crew analysis.

The technology challenges that exist include executing the rollback of the software update to the prior version without disruption to spacecraft control, determining when to roll back to the prior version, and determining the rollback sequence. The ability of the software to decide when to accept or not accept a software update will reduce required interventions by the ground to troubleshoot and correct errors in new software loads.

3.8.6 Development Plan

The development plan for software autonomy needs consists of architectural and data-centric technologies. Data is central to providing the situational awareness to understand, operate, and maintain the spacecraft, whether via human control or the control of autonomous technologies. As such, many of the technology developments are focused on data. The other architectural developments that will be covered include considerations for command authority, configuration management, and software updating.

3.8.6.1 Architecture Developments for Software Autonomy

3.8.6.1.1 Data Selection and Management

From the initial design of the software concept of operations, the understanding of what data will be needed for the functionalities that are desired is essential. Data selection will drive sensor design, selection, number, and placement. Data management will dictate storage and processing architecture and hardware. Data is not typically the first thing that an engineer considers, however, when designing a spacecraft subsystem. Design philosophies and methodologies must be developed and adopted that encourage a model- and data-centric view of the system to ensure that the appropriate hardware architecture is available to support the software and autonomy needs of the system. While industry is likely to push these sorts of architectures and design methodologies for complex data-centric systems, NASA must continue to invest in partnerships and training opportunities for systems engineers to become experts in these technologies.

3.8.6.1.2 Configuration Data-Driven Software Engines

Most, if not all, of the decisions for system and subsystem monitoring and control are based on the actual values in the operational data. Portions of the decisional control will rely on configurable data to interpret or act on the operational data. As a result, the executable software becomes a data processor or “engine” whose operation is largely dependent on and driven by the value(s) of the various types of data it handles. Building a configurable data system in this way allows for the modification and validation of data while minimizing flight software modifications. Although the data-centric approach has been used with positive results in prior projects, NASA would benefit from continued investment in methods and techniques that mature these configuration data technologies on a large-scale.

3.8.6.1.3 Distributed Autonomy Architectures

While a clear need for a VSM exists, this centralized system must be able to work with subsystem software (which also may have hierarchical software functions). An architecture that allows for the seamless sharing of data and control across subsystem boundaries (and hierarchical software boundaries) is important. For spacecraft applications, duplication of data storage or processing across software functions (when not done for criticality or redundancy reasons) carries a potentially unacceptable cost. As such, the development of technologies that allow distributed handling of FDIR, maintenance, and control, while respecting the highly integrated nature of the spacecraft, will be important to finding the optimal autonomy solution to the challenges that will need to be resolved without outside support during uncrewed periods. Examples of technology developments needed for this could include efficient communication across autonomy components, variable distribution of autonomous control across components, crew, and ground control, models with embedded abstraction levels, hierarchical task planning, and assume-guarantee contracts across autonomy elements.

3.8.6.1.4 Software Updating Framework

Software updates are expected during the long dormant period in this design reference mission. Since ground controllers will have limited communications, it will be important that the software updating scripts have the needed intelligence and autonomy to determine the status of the update and to take the necessary steps to resolve any problems. This problem is common in industry, and best practices should be taken from software updates in critical applications and applied to this problem. While NASA investment into developing these technologies is not needed, concerted effort to ensure that these best practices are incorporated into the software architecture from early design stages is essential to the success of this function.

3.8.6.1.5 Configuration Management

A clear understanding and control of the configuration of all the software on board the spacecraft is essential to maintaining situational awareness of the remote spacecraft. Similar to the software updating technology, configuration management best practices for critical systems are being developed in industry presently. NASA must stay abreast of these technologies and plan to adapt them to the specific use case of the human spacecraft. Configuration management choices will affect the overall software architecture, and should be considered early in the design process.

3.8.6.2 Data-Centric Developments for Software Autonomy

3.8.6.2.1 Intelligent Data for Procedure Execution

In complex integrated systems, it is common to have a configuration change or control command in one part of the system drive unintended effects in other parts of the system. When procedures that consist of these configuration changes and control actions are being autonomously executed, not only will expected data need to be available to the procedure execution engine, but associated events that are unexpected must also be identified, captured, and available. Because these interactions are, by definition, unexpected, the data itself must be collected in such a way that possible associations will be explored. This may require annotations, logging and storage methods, and deep analysis functions in order to capture the unexpected consequences.

3.8.6.2.2 Data for Learning Systems

Learning and adapting will be important for the autonomous functions on the uncrewed spacecraft, particularly since it will be operated in a relatively unknown environment. Currently, significant human time investment is required to collect and annotate data required for learning. On the dormant spacecraft, this will be difficult to accomplish due to the bandwidth limitations to Earth, no crew on board, and limited surface crew time available to service the on orbit spacecraft. As such, technologies must be developed to enable the autonomous annotation and classification of data for learning systems. While this seems like a circular problem, promising work has been explored in this area. Further developments are needed to push the TRL of this idea.

3.8.6.2.3 Variable Content Data Downlink

Because of the large amount of data that will be expected to be generated from the spacecraft, even during the uncrewed period, and because of the bandwidth restrictions that are expected from Mars (considering also that communication with the crew on the surface of Mars will be ongoing during the same time period), it is essential that the correct, prioritized data is downlinked from the spacecraft. While this is a technology that has been described for several subsystems, the point to be made here is that the data itself must contain annotations or be organized appropriately in case downlink conditions change unexpectedly (i.e., more data needed from the crew on Mars, degraded communications capability, etc.). As such, developments for the organization and annotation of data for this autonomous prioritization and downlink capability is needed, and is clearly an investment that NASA should make due to the specific use case.

3.8.6.2.4 System Robustness to Incomplete or Corrupt Data

Due to the extreme environment, it is likely that data will be missing or corrupted at some point during the mission. Since the point has been made that data is essential to the operation of many of the autonomous spacecraft functions, special consideration must be given to designing the system to handle off-nominal data conditions. While systems developed in industry must handle similar considerations, this design reference mission likely provides a bounding use case (or an edge case). As such, NASA investment into system robustness guarantees to an expected frequency of lost or corrupt data is needed.

3.8.7 Criticality and Redundancy

3.8.7.1 Software Criticality

All flight software is typically developed as Class A, Safety Critical products. Very few exceptions exist if the software controls any aspect the spacecraft since the software ultimately supports the crew. When a software product is in question regarding safety criticality, the decision defaults to the highest criticality level as it is cheaper to develop software at that higher level of criticality than it is to certify to the higher level later.

During dormant operations, software criticality is tightly coupled with the criticality of the subsystem in which it is embedded. Software always assumes this state as the enabling logic that controls the subsystem, device, or component. This being said, the software will not transform itself into a less critical configuration.

Software that is not embedded in a subsystem, but controls any aspect of the spacecraft such as the VSM is by definition Class A, Safety Critical. Even during dormant operation with no crew onboard, the VSM is either maintaining an environment that the crew has to survive in or is capable of returning the environment to a habitable state.

Standalone systems that interact with the crew and/or the spacecraft such as robotics would typically develop software as Class A, Safety Critical. Due to the human interaction component, the software in these systems would assume the same level of criticality as the system in that mission phase.

Payloads, experiments, and other devices or instruments not integrated with the spacecraft flight processing systems usually follow their own criticality assessment. This is based on safety classification assessment as defined in NPR 7150.2B. Generally, the criticality assessment will not change based on the dormant phase. If data is transferred to the primary flight systems for storage or telemetry, that system will be designed in such a way as to cause no harm if it were to fail.

In general, software criticality is tied to the criticality of the system or subsystem, which in turn is tied to the flight phase or mission objectives in the dormant S/C. This is the nature of embedded software.

3.8.7.2 Software Redundancy

Two types of redundant software systems can be defined, active and passive. Active is duplicate software executing on a “back up” processor in a redundant subsystem or component. Passive is the retention of a duplicate copy of the executable software and data in storage, which is not susceptible to SEUs or corruption.

Generally, software redundancy is tied to the system’s or subsystem’s criticality definition and fault tolerance requirements. If for example the spacecraft is designed as single fault tolerant, a duplicate, active instance of the software will run on a backup processor. As such, these systems would generally use fail-over schemes to engage the secondary or back up system if the primary were to experience faults or failures. This concept can be extended to multiple flight processors, however this implies voting systems, which typically imply the need for higher fault tolerant levels and levels of complexity

Cross-strapping is another option that can be traded based on subsystem criticality. Such a configuration provides the subsystem with the ability to share assets with its redundant string if failures eliminate some or all of the primary system.

For systems that are designed to be recoverable, duplicate images of the executable software are passively stored in hardened or non-volatile memory. If the executable image or data is corrupted in main memory, the software can be reloaded from this protected memory based on fault detection and recovery procedures.

Ultimately software redundancy is dependent on system and subsystem criticality. For primary flight computers, the idea of single fault tolerance will drive the need for a primary/backup scheme that can be managed by the flight software. It would be advantageous to use a combination of active and passive redundancy on stand-alone flight computers and processors in most flight subsystems.

3.9 Spacecraft Emergency Responses

Traditionally, Spacecraft Emergency Responses are built into the ECLSS. 'Emergency' for human-rated vehicles is defined as an event that poses an immediate threat to crew safety. Using this definition, Spacecraft Emergencies usually fall within one of three categories: fire, cabin depressurization, or atmosphere contamination via toxic release. Because each of the three categories of emergency events affect cabin air or cabin pressure, spacecraft emergency response is highly reliant on ECLSS for insight and failure response.

3.9.1 Nominal Operations

When the spacecraft is in a dormancy period, the lack of crew onboard the vehicle eliminates immediate life-threatening emergency scenarios, which greatly increases time to effect for the associated failures. The vehicle configuration for dormancy should mitigate the risks and effects of emergency scenarios as much as possible. Non-essential equipment, especially equipment with moving parts subject to overcurrent events or friction-seizing, should be powered down to mitigate risk of fire. Hatchways between modules should be closed to mitigate the spreading the effects of fire, depressurization and toxic release, reducing the volume of air that would need to be scrubbed should one of those events occur while the spacecraft is dormant. Once in a dormant state, any of the ECLS Systems that are powered and able to monitor for emergency conditions will continue to do so, providing the remote crew and ground with insight and the ability to trigger post-emergency clean-up as required. The spacecraft design must include air purification and resupply capabilities that can be operated in the absence of the crew, in order to restore a safe environment following an emergency event prior to crew return.

3.9.2 Transition Operations

The transition phase of emergency response systems and processes on the spacecraft will focus on the preparation and changes to response methods that are needed for crew arrival and their departure.

3.9.2.1 Transitioning from Crewed Operations to Dormancy

The transition to dormant operations for emergency response systems will center on enabling a fully autonomous detection and isolation system. The focus of emergency response during the dormant phase will prioritize safety and preservation of the vehicle. To mitigate damage to the vehicle, quick automatic response to isolate emergency conditions to confined areas followed by autonomous corrective and recovery actions are assumed to be available for dormant operations scenarios. Non-critical systems should be deactivated for dormancy periods to mitigate risk of emergency scenarios due to system failures. Any features of the emergency response system that were disabled during crew habitation should be enabled upon their departure. The action of enabling the emergency response system for autonomous operations should be available onboard and from ground support to allow flexibility in commanding.

3.9.2.2 Transitioning from Dormancy to Crewed Operations

Emergency response methods with crew present on the spacecraft are assumed to incorporate crew assistance in the detection, isolation, and recovery process. The focus of emergency response during the crewed operations phase will prioritize safety and preservation of the crew. While some autonomous detection and isolation techniques independent of crew action may remain enabled with crew present, any automatic response that could jeopardize crew safety or crew ability to retreat to safe haven must be inhibited or modified during crew habitation periods. Additionally, crew presence may also require additional system monitoring capability for hazardous conditions as crew support systems return to a fully operational state.

3.9.3 Contingency Operations

Emergency response systems and processes seek to address failure events that jeopardize the safety of the crew, continued operation of the vehicle, or the ability to complete the mission. Emergencies on a spacecraft are categorized into three types: fire, rapid depressurization, and toxic atmosphere. During dormant phases of the mission, when crew is not present, the focus of emergency response systems is on maintaining minimum vehicle system function in order to support crew return. During transition to and from dormant operations, when crew would be present, the focus of emergency response systems shifts to maintaining minimum vehicle system function in order to support crew habitability. While a Fail-Operational state is a good target during emergency response scenarios, the severity of the emergency may drive the response to a Fail-Safe state.

In order to support dormant spacecraft operations without crew present, all emergency response systems (fire, depress, toxic release) should be telemetered with basic sensor checking included in the CDH system such that manual (human) verification of system health is not required. Emergency scenarios can escalate to catastrophic levels quickly, resulting in an inability to maintain a Fail-Safe state if manual verification and response is solely relied upon.

3.9.3.1 Fire

For a dormant spacecraft without crew present, the fire suppression system should be designed to operate without crew intervention, starting with design for minimum risk (as was done on ISS) and then adding in-situ fire suppression capability at points of greatest risk. During the transition to and from crewed operation, automatic fire suppression may still be desired, but consideration should be given to the effect of that suppression on crew who may still be in the cabin. For example, a CO_2 fire suppression system can create a toxic environment.

To achieve automated fire/smoke response, a reasonable approach may be to power down the most likely causes of fire – high current equipment or equipment with moving parts where friction or resistance could be contributing factors. That said, operationally one finds that assumptions made in the design phase tend to not work precisely by the time the vehicle is operational, so a “power-down approach” includes a risk that the emergency response system will not be designed to respond to failures in some systems. This gap in emergency response capability results from the assumption that equipment would be powered off during initial design, but in later iterations the systems will need to be powered on.

Consideration must also be made for the relationship between sensors and environment conditions. For smoke sensing devices such as those in use on ISS to work properly, for example, air ventilation equipment must continue to operate in dormancy periods. However, the fans that provide air ventilation contain moving parts with potential to cause a fire themselves.

3.9.3.2 Rapid Depressurization

Emergency response to a rapid depressurization event must address the detection of the leak location, the ability to isolate the identified leak area, and the recovery of attitude control during or after the venting disturbance. The speed at which the leak location can be identified and isolated will directly relate to the amount of atmosphere lost, which then drives the disturbance force on the vehicle. For a dormant spacecraft without crew present, an automated isolation capability should be provided to minimize atmosphere loss or the spacecraft should be left in an isolated configuration when transitioning from crewed to dormant operation. An isolated configuration includes closing of hatchways or other methods to compartmentalize the interior volume of the spacecraft in order to minimize atmosphere loss due to a hull puncture.

When transitioning into and out of a dormant phase configuration, isolation methods must ensure the crew always has a safe evacuation path as compartments are closed or re-opened. To accommodate crew egress, additional atmosphere reserves must be available to maintain pressure at safe levels.

Restoration of breathable atmosphere may first require repair of the spacecraft hull before habitation by the crew. Design of the spacecraft structure should address the ability to identify very small punctures in the pressure shell and conduct repairs both from within or outside of the hull, with both cases involving EMU suited crew and/or robotics capable of performing the repair in a vacuum environment. This ability would provide for repair options while minimizing the expense of atmosphere to feed the leak during repair. Additional complication with re-ingress to a depressurized section exists in this case, depending on location of airlock relative to the depressurized section and the size of hatchways relative to the size of a suited crew member.

3.9.3.3 Toxic Atmosphere

Toxic atmosphere is generally defined relative to toxicity for humans. For a dormant spacecraft without crew present, there would be no impact of toxicity but the substance could be corrosive or damaging to internal vehicle hardware. Air quality and filtration of the atmosphere, already addressed in ECLSS subsystems, should be capable of sensing and removing toxic contaminants. Replacement of filters, or regeneration of the filtering ability, should be addressed in the context of response to a toxic atmosphere event. If the filtering of atmosphere to remove the toxic contaminant must occur at the moment of detection, then filtering capacity must account for removal of the toxic substance as well as continued function to address possible recurrence of the emergency. Alternatively, filtering could be postponed to occur just prior to crew arrival and transition to crewed operation in order for the arriving crew members to address filter replacement or regeneration.

3.9.4 Preventative Maintenance and Logistics Support

Sensors involved with the active monitoring for emergency scenarios on the spacecraft should be subjected to diagnostic review to verify consistent operation. Depending on the sensor type and complexity, the diagnostic review may itself be an automated computer task by a connected asset or may require self-diagnostic or self-repair by the sensor itself. For optical detection systems, cleaning of the optics without crew intervention should be accounted for in the sensor design during the dormant phase.

The ability to replace sensors onboard should be considered in terms of the capability to conduct a replacement operation via robotics. If the sensor cannot be readily replaced with onboard assets (spares available, robotics capability), then a higher redundancy requirement should be imposed on the sensor to reduce reliance on replacement needs.

Consumables available for use in a response to an emergency should be sized appropriately to accommodate a worst case scenario derived from the duration of the dormancy period. For example, a long dormancy period may warrant a higher consumable budget for emergency response due to the longer exposure window. Resupply of the consumables for emergency response could then be planned for the next crew (or resupply) mission depending on if the onboard supplies have been used for an event, or have expired.

3.9.5 Autonomous Functions Needed

The current state-of-the-art in emergency response systems for a spacecraft relies on a blend of onboard crew and automated system responses to quickly address emergency scenarios. Readiness to react to emergency scenarios is maintained through consistent inspections of sensors and response equipment. During an emergency scenario on the ISS, the crew executes the actions necessary to first provide for their own safety, then to isolate the emergency event, and finally, to repair the damage when possible. Automated processes onboard and ground commanding are used in parallel to crew actions on ISS to aid in event isolation and recovery, increasing crew efficiency during the response period.

For a dormant spacecraft without crew present, autonomy should be applied to address the actions that had been allotted to the crew when onboard when those actions are needed for vehicle preservation. In cases where time to criticality does not support ground commanding responses, the autonomy should include the ground actions in addition to crew actions. In most instances, ground actions have to do with providing situation assessment information and post-event clean-up actions. Human spaceflight missions to date have relied on crew actions for emergency response to ensure no reliance on the ground since the ground loses communication with the vehicle in the normal course of operations. While the absence of crew onboard may extend the time to criticality for emergencies, onboard systems will be impacted, resulting in a need for action.

3.9.5.1 Fire Emergency Response Autonomy Needs for a Dormant Spacecraft

Detection of a fire emergency through the use of smoke detectors and confirmed by onboard crew members for current spacecraft has proven effective. However, the current design of smoke detectors is susceptible to false indications of fire due to dust in the atmosphere. Onboard crew members must confirm the presence of smoke or fire in a fire event. There is no current implementation for fire detection that couples the smoke detector signature to subsystem performance or other sensors in order to identify the source of the fire without localized atmosphere constituency measurements (which require crew on the ISS). Advancements in fire location will lead to more efficient fire response, resulting in reduced need for widespread power-downs and more accurate deployment of countermeasures.

Countermeasures for fire scenarios must be capable of deployment without crew intervention during dormant operations. Reliance on powering down equipment alone may not be sufficient to inhibit the spread of the fire to other equipment. Deployment of countermeasures should also account for consequential impact to neighboring subsystems and atmosphere, as the module will need to be reingressed at some point to conduct equipment repair. Automated shutdown of airflow during fire emergencies already exists for the ISS, however, the hatch closures are dependent on crew action. Consideration should be given to module or regional isolation to minimize atmosphere contamination from smoke and countermeasures. Air filtration capability should also be addressed and sized to accommodate recovery of the atmosphere after a fire response event, while also preserving margin for nominal operation after crew return.

3.9.5.2 Toxic Atmosphere Response Autonomy Needs for a Dormant Spacecraft

Detection of a toxic atmosphere event requires both air quality sensing and sensing within any systems that pose a threat of toxin release in a failure scenario. Toxic atmosphere can be caused in a variety of ways even without the presence of toxic substances in major operating systems. Plastics used in component housings can off-gas toxins if those components overheat. Fire suppression systems using CO_2 create a toxic air environment that must be scrubbed. Corrosives like traditional batteries can create toxic particulate if they come into contact with unprotected materials.

On the ISS, the primary toxic atmosphere event addressed in crew and ground training and onboard automation involved the release of external cooling fluid into internal compartments via a failure at a heat exchanger. Most coolants with properties that are suitable for space applications are highly toxic to humans and system design should include sufficient containment mechanisms to prevent atmospheric contamination as a result of a thermal system failure.

Each type of toxic event must be evaluated to determine whether cleanup response is required during the dormancy phase, to protect vehicle hardware, or required prior to crew return.

The isolation process itself should be capable of preventing the spread of the toxic atmosphere, while also preserving the ability to clean the atmosphere where possible. Recovery of the atmosphere will reduce gas resource needs. However, if overboard venting of atmosphere is required to clear the toxic contamination, consideration should be given to the integrated impacts of such venting on vehicle attitude control, repressurization capability, potential contamination of interior surfaces, and operation of neighboring C&DH, Avionics, and Power systems during the depressurization period.

3.9.5.3 Rapid Depressurization Response Autonomy Needs for a Dormant Spacecraft

Shielding and debris avoidance are two proven methods to avoid a rapid depressurization event on a spacecraft. Shielding and debris tracking capability, if possible, should be implemented in tandem to ensure no gaps in particle size between what the shielding can protect against and what size particle can be tracked and subsequently avoided through orbit adjustment. Any gap in capability to address particle size will result in significant reliance on the speed of response to successfully safe the vehicle and isolate the hull breach.

Detection of an atmosphere leak resulting in a rapid depressurization event will be critical to enabling quick isolation of the ruptured hull area, potentially reducing the loss of gas resources onboard the spacecraft. On the ISS, the detection of an atmosphere leak can be triggered by pressure sensor readings, but presents a challenge to determining precise leak location without onboard crew response. Advancements in hull integrity sensors should be pursued to identify the specific location of breaches rapidly. Both large and small leak cases should be considered for the dormant spacecraft, as a small leak can have significant impact to reserve gas stores on the spacecraft, as well as detrimental impacts to attitude control efficiency during the dormant phase.

For a spacecraft such as the ISS, which consists of several modules with inter-module hatchways, the capability to quickly close hatchways will reduce the amount of air loss and the duration of the vent force. Analysis of likely leak scenarios for ISS indicates that reducing the vent force magnitude and duration will mitigate the impact to attitude control capability, as well as subsystems that rely on pointing direction such as Communications, Power, and Thermal systems.

3.9.6 Development Plan

The most effective option for bridging the autonomy gap in emergency response would be to design the overall system such that the emergency cannot occur. This requires dedicated system engineering, and may be restricted by constraints imposed by other necessary functions of the spacecraft system. Other methods of the avoidance of an emergency, such as developing a system that generates Debris Avoidance Maneuvers may be exceptionally difficult given the lack of infrastructure that exists in Mars orbit or due to other factors. However, these emergency avoidance techniques, whether due to system design or due to mitigating procedures or technologies, merit investment and further development.

When removing the potential for the emergency is not possible, the emergency response subsystem function is required. The major development theme within this subsystem centers around a system trade between distributed sensing and actuation and robotic agents. In all cases, the determination and the localization of the emergency is the first step, followed by isolation, and finally recovery. Discussion on the technology development needed for each of these three stages will follow.

3.9.6.1 Determination and Localization

The crew acts as mobile sensor packages on board the ISS. There are two main ways to replace the crew in this function. First, networks of sensors deployed in smartly designed ways can be used to identify and localize fires and leaks. Considerations about the design of sensor networks is covered in Section 3.13.1. Developments in the creation of sensors that can effectively identify more than one type of emergency, in sensor size, mass, and power consumption, and in robustness to failures are required for this path to be feasible. Failures in distributed sensing must be accommodated via the overall design, such as in the robustness of the sensors to failure, the distribution function of the sensors being designed to accommodate a certain percentage of failures without affecting overall performance, the ability to replace sensors via robotic means, or some combination of these things. Sensor failures must also be accommodated by understanding the signature of these failures. Particularly for emergency responses, false positives due to sensor failures are extremely costly (wasteful of resources and/or harmful) and must be avoided. Power, data logging and storage, and computation resources must also be allocated to this solution, as powering and accessing the data from these sensors will have a cost. Innovative methods for utilizing only a percentage of the sensor array until an event is suspected could be developed.

Further developments in sensor fusion and virtual sensors could also benefit the emergency response function. By using many different types of sensors and fusing the data together, it may be possible to determine likelihood of an emergency faster than using dedicated sensors. An example would be using trace contaminant sensors in conjunction with sensors that measure particulate content of the air to reason about potential fire conditions. Virtual sensors may also contribute to smaller sensor networks if model inferences between many sensors are able to detect events without having co-located sensors. The emergency response system has many indications that are possible to sense to determine if an emergency has occurred (or is occurring). As such, research into what the minimal set of sensors could be would be a sound investment.

The second path towards replacing the crew member involvement in emergency scenario identification would be to deploy robotic mobile sensor packages. Instead of arrays of sensors, mobile sentries carrying these specialized sensors could be used to confirm and localize emergencies. These mobile robots could be alerted to suspected emergencies and deployed to further investigate, or act as a sentry to look for incipient stages of problems, if possible. For example, thermal imaging could be continuously conducted by the robotic agent, looking for hot spots in response to contaminant sensing. The robotic agent could carry many types of sensors, which could be more massive and otherwise expensive than a networked version. However, the added mass and complexity of the robot would have to be considered. Technology development for the mobile inspection robots will be outlined in Section 3.7.6. However, specifically for emergency response robot inspections, research into the optimal set of sensors to carry is needed, similar to the work that is needed for sensor arrays. Another challenge introduced by mobile sensors is the appropriate way to fuse the mobile data with the sensor array data in a dynamically consistent way. This is something that should be investigated.

NASA should invest in both options, with a planned trade study at an appropriate point that determines the costs and benefits of each of these methods, or some combination of both.

3.9.6.2 Isolation

Once the emergency has been identified and localized, it must be isolated. System design is essential to determining what sorts of isolation mechanisms will be used. Isolation is important to autonomy because, when it works, it increases the time to criticality of the emergency, which reduces the complexity of the autonomous response. Hatches are currently the primary method for isolation on board the ISS. Other methods for isolation should be considered within the system design for new vehicles, given the constraints in mass and power. Investments in isolation method concepts that are more effective should be considered, such as a fire suppression system, actuated valves (either self or robotic), or temporary patches. If hatches are the chosen isolation method, that design must trade the costs and benefits of a distributed sensor system versus a mobile sensing system since a mobile sensing system would likely be inhibited by a hatch-driven isolation design. Alternately, hatches could be designed to be self- or robotically-actuated while maintaining a manual override capability to preserve mobile sensing options. The software for self-closing hatches would need to be fully verified against any potential of causing crew members harm. This could be accomplished by provably turning off the ability to close autonomously while crew members are on board, or by developing an intricate fault-tolerant control system.

3.9.6.3 Recovery

Recovery from emergency responses could take many forms. For all recoveries, an assessment of the situation is required. For that, sensor arrays or mobile sensors must be capable of providing enough situational awareness for ground controllers or autonomous algorithms to ensure that the appropriate actions are planned and undertaken. Studies and research into sensors similar to the efforts needed for determination and localization are required, as are developments into sensor registration algorithms. In particular for this situation, which may have a longer time constant for operation, but that involves delivering a lot of unstructured information to the ground controllers, technologies for expressing situational awareness over low bandwidth and high latency connections are essential. NASA investment is crucial for this technology development, though commonalities could be found in disaster relief scenarios on Earth.

Recoveries from deployment of the fire suppression system will include clean-up of atmospheric contaminants and possibly repair or replacement of affected components. Mobile robotic manipulators are the best candidates for repair/replacement efforts, if the recovery must be done during the uncrewed period due to criticality of the affected subsystems. The technologies needed for this are detailed in Section 3.7.6.

Recovery of the atmosphere from a toxic leak event could include filtration or venting, and systems should be designed for the appropriate recovery capability. For time critical events, it could be necessary that the autonomous system would need to determine the appropriate actions to take to recover the atmosphere. In that case, planning technologies that respect constraints and dynamics would be required, and investment is needed for the advancement of these.

Recovery from atmospheric leaks, either large or small, could be handled by either self-healing structures or the robotic ability to access and repair hull breeches. Investments in materials that are light, strong, and have the ability to mend are recommended for NASA. Developments that would be required to have the robotic ability to repair leaks are detailed in Section 3.7.6.

3.9.7 Criticality and Redundancy

Emergency response systems are considered level 1 criticality and should require two fault tolerance for sensing an emergency event and the response equipment. The spacecraft systems must also be considered when addressing safe operating environments that prevent causes for emergency events, such as fire.

Placement and size of the sensor network should consider the dormant spacecraft phase as a primary sizing driver. During crewed periods, the crew's senses can serve to supplement spacecraft sensors, resulting in a shared reliance on crew and sensors to identify emergency scenarios. However, with crew absent, sensors must be dependable (achieved through design and redundancy) and placed in such a way to provide quick localization of a problem, which can then lead to a quick response and targeted safing actions. Previous crewed spacecraft design reliance on crew identification of emergency events should not be under-estimated when designing for a dormant spacecraft capability.

Equipment used in the response to an emergency should dependably operate autonomously during dormant phases. Fault tolerance and sufficient consumables to address emergency scenarios should be included in the system design. As the vehicle transitions to support crew return, the capability should exist to reduce the autonomous response capability, as crew response options may need to take a higher priority.

3.10 Structures

The basic function of the spacecraft structural system is the support of all of the integrated subsystems of the spacecraft. The support includes the safe containment and storage of all of the required fluids on the spacecraft in liquid or gaseous state. The total compliment of fluids includes the spacecraft breathing air atmosphere in the pressurized crew compartments.

The structural certification of all spacecraft subsystems is performed prior to launch and assembly of all elements to certify structural integrity of all spacecraft structure and subsystems for all of the imposed loading conditions throughout the fifteen-year design life of the structure.

The structural certification is performed to include all imposed structural loading conditions for the crewed operations and dormancy periods, including load cases that are applied prior to launch of the spacecraft.

3.10.1 Nominal Operations

The spacecraft structural system will require SHM instrumentation and data recording capability. The SHM should include distributed loads and dynamics instrumentation, habitat module impact detection instrumentation, and habitat module atmospheric leak detection and location instrumentation. The system has to monitor on a continuous basis, however data reporting is only required on demand or on the occurrence of data values that exceed limit load cases. The SHM does not include effectors or capabilities for recovery operations, such as module isolation and repress.

3.10.2 Transition Operations

Since SHM is always active in both crewed and dormant periods, no transition operations are necessary.

3.10.3 Contingency Operations

There are two types of contingencies for the spacecraft structure. The first is a high load event without an associated leak. The second is an atmospheric leak from any of the pressurized modules of the spacecraft. For the high load event, likely precautionary operations would include inspection of the area experiencing the load and possibly additional data collection.

An atmospheric leak may be caused by various potential events, such as micro meteor impacts, collisions with other spacecraft or robotic assets, operational damage, equipment failure (valves, seals, payloads, structural failure), and corrosion or erosion causing a metal degradation process resulting in loss of pressure integrity. The atmospheric leak may be very low flow rate or may be a very high flow rate, depending on the equivalent hole size through which the atmospheric flow is emitted. Responses during crewed periods may be expressed in terms the response time available down to 10.2 PSI cabin pressure, which is the hypoxia level. It is expressed as T_{res}. The minimum acceptable value is 15 minutes. This is assumed to be enough time to transit to an escape vehicle and depart the spacecraft. For dormant times, the responses may be expressed in terms of the pressures required to maintain critical functions and processes, which may be lower than the hypoxia level.

The leak response has the following requirements. First, there must be instrumentation in the spacecraft to find the atmospheric leak source (i.e., which pressurized element is leaking and the coordinates of the leak within the element), a leak alarm must sound in the pressurized volume for the crew, and automatic computation of the leak rate based on the pressure drop to establish available T_{res} for response must occur. During the dormancy period, all hatches should be in a closed position, if possible, to prevent total depressurization in case of a single module leak. Finally, the leak will need to be repaired if possible. The leak could be repaired IVA if there is full IVA access to the pressure wall and an IVA pressure wall repair kit. If not, the leak would need to be repaired via EVA.

3.10.4 Autonomous Functions Needed

The SHM system has three main autonomy gaps, two that relate to how often the data must be viewed by ground control personnel and a third that relates to the robustness and reliability of the sensor network.

3.10.4.1 Event Detection and Smart Logging

Since the SHM is responsible for capturing the data required to determine if an excess loading event has occurred anywhere on the spacecraft structure, the system becomes more independent from ground oversight if it can detect these events autonomously and adjust its logging prescription to suit. For example, if the SHM identifies an event, it would autonomously keep high rate data that corresponds to the event and flag it for immediate downlink to ground controllers. This autonomous functionality would also need to be verified to ensure that all relevant data was captured for every event in order to provide the trust necessary in the system. The event detection would also allow the system the ability to reason about what has happened to the spacecraft overall, which would enable emergency response actions if necessary.

3.10.4.2 Localization of Load Events

Supplemental to the event detection is the ability of the SHM to localize the load event and, where possible, to correctly assess or interpret local events, for example, identifying serious threats versus benign events. Localization is something that currently requires some human input, either by looking at the data or by crew inspection. In order to increase the amount of time that can pass between human observation of the system, the SHM should be able to provide the vehicle management system a determination of where the event occurred in order to inform the next steps as needed. Furthermore, autonomous assessment may provide the crew members or ground controllers with better and faster threat information, which could be invaluable in case of a life or mission critical threat.

3.10.4.3 Sensor Network Robustness

Increasing the robustness of the sensor network would fill the final autonomy gap for the SHM system. This could be accomplished by reducing the downtime of the system, providing heartbeats and other health checks of individual subsystem components and sensors, and by sensor design itself.

3.10.5 Development Plan

The SHM development plan is proposed to be based on ISS SHM development. Likewise, the sensor network development is to be based on the design of the ISS sensor network for SHM. Unique autonomous algorithm development advances are required for event detection, characterization, and assessment, in-situ data analysis, and the V&V of autonomous systems.

3.10.5.1 Systems Engineering Design for Structural Health Monitoring

The sensor network for SHM should include:

- distributed loads & dynamics instrumentation;
- spacecraft and mission critical hardware meteoroid impact detection instrumentation;
- spacecraft atmospheric leak detection and location instrumentation; and
- ECLSS and active thermal control system (ATCS) fluid leak detection and location instrumentation.

The sensor network development for the DSG SHM evolves from the existing ISS SHM solutions such as the Loads & Dynamics data acquisition systems (internal and external wireless instrumentation systems) and the Cabin atmosphere and Fluid Systems Leak Detection and Location Systems (Atmospheric

Pressure Wall Leak Location System and Ultrasonic Background Noise Test). The common challenges of designing sensor networks are treated in Section 3.13.1.

Research into materials and configurations that may naturally support SHM, such as inflatables and self-healing structures is an interesting system design direction that should be explored. The game changing potential that these technologies provide are worth the investment into the low TRL concepts.

3.10.5.2 Dedicated Algorithm Development for SHM Autonomous Systems

3.10.5.2.1 Event Detection, Characterization, and Assessment

Similar to fault detection, this triad of functions creates a robust process to determine when a set of data has indicated that an out of limit condition is reached. Event detection and the subsequent characterization may use an existing model or be a learned model (or a combination of both). However, the algorithm developments are needed to assure that event impacts on structural integrity are understood and events are not missed. It is also required that the algorithms minimize or exclude false positives (to avoid the cost of the action plan that results from a mischaracterization of an event). Event detection development to date has involved fairly simple cases (such as fault detection when the signature is modeled and expected). Most often, event detection for SHM of spacecraft beyond low Earth orbit will operate on modeled or expected event signatures, but these missions are expected to record a critical subset of un-modeled and unexpected events. Similar research in other applications is ongoing for algorithms and systems that can identify when unexpected events occur and assess the situation (for example - spontaneous science detection for robotic planetary explorers). However, continued NASA investment is essential for developing the technologies and algorithms that are needed to determine when an unexpected SHM event has occurred and provide the critical structural assessments of the impact on mission success.

An option for event characterization and assessment includes mobile sensor and manipulation platforms. These robots can be invaluable in providing data needed to fully understand what may have happened during a detected event. A more complete treatment of the use of robotics for this function can be found in Section 3.9.6.

3.10.5.2.2 In Situ Data Analysis

In-situ data analysis technology applies to more than just SHM. However, the process of determining causation, localization, and structural assessment from a sensor array's operational data is a difficult task (given the complexity of the structure, the possible amount of data, the environmental unknowns, and the relative lack of situational awareness). The SHM application requires the development of low and high fidelity structural static and dynamic models as well as the system's response models to represent the actual hardware. These models will be at the heart of the functional autonomous assessment and characterization algorithms. Also, these models will require the development and ground truth testing to insure their quantitative value, including assessment of the environmental baseline conditions. Additionally, predictive algorithms which are able to assess damage states resulting from impacts and system failures will need to be developed. The development and implementation of advanced learning systems may also be required to satisfy long-term requirements. The spacecraft's autonomous systems must be able to provide answers using the processing capabilities resident on board, even while such answers involve solving complex mathematical problems. NASA investments in creating data analysis solutions that can accommodate limited flight resources are a valuable and necessary investment.

3.10.5.2.3 Verification and Validation of Autonomous Systems

The assurances required of these autonomous functionalities to gain the trust of ground controllers derive from the V&V of the developed systems. Likewise, these SHM capabilities will be required to accommodate systems that change (such as sensor networks that have various configurations as individual sensors degrade and fail) or models that adapt for environment unknowns (and reset to an updated nominal state). This technology is discussed in Section 3.13.4.

3.10.6 Criticality and Redundancy

The Criticality category recommendation for the SHM is 2R. Where redundant sensor arrays within SHM hazard monitoring system exist, if all of them fail to detect or operate when needed during the existence of a hazardous condition, it could lead to loss of the crew or the spacecraft.

SHM sensor arrays are comprised of redundant multiple sensors. The array architecture requires redundant data processing and data collection means, where failure of portion of the array architecture will enable a degraded SHM function to provide sufficient resolution to provide state data to the crew and ground controllers.

3.11 Thermal

All spacecraft require a thermal management system to maintain a tolerable thermal environment for the spacecraft crew and/or equipment. For the purpose of dormancy activities, as described in the present work, the role of the active thermal control system (TCS) is to maintain the cabin environment and vehicle systems within acceptable temperature ranges. The requirements for human rating and the specified controlled temperature range (approximately 275–310 K) for crewed spacecraft are unique, and key design criteria stem from overall vehicle and operational or programmatic considerations. These criteria include high reliability, low mass, minimal power requirements, reasonable development and operational costs, and high confidence for mission success and safety. A number of references are available to describe spacecraft thermal control in detail, including *Hurlbert, K. M., "Spacecraft Thermal Management,"* in *Encyclopedia of Aerospace Engineering, R. Blockley and W. Shyy (eds.), John Wiley & Sons Ltd, Chichester, UK, pp. 481-492, 2010.*

3.11.1 Nominal Operations

Nominal operations of the TCS during dormancy would include primarily monitoring of instruments and parameters, and automated analyses of the data to determine if the parameters remain within acceptable ranges. It is assumed that there will be flowing fluids in both the internal and external coolant loops, with active (although reduced) levels of heat transport and rejection. Operations would also include any adjustments (e.g., mixture, loop volume, flow path) via components such as pumps and valves.

The most critical functions during dormancy are monitoring instruments and parameters, and automated analyses to determine if any control functions or adjustments are needed. There will be numerous sensors and parameters to be monitored and automatically evaluated (e.g., temperatures, pressures). Should the automated data collection and associated evaluations demonstrate certain parametric ranges, then the system should automatically make adjustments to the mechanical device(s) (e.g., pumps) required. There should also be pre-programmed lag times and/or evaluation cycles to allow sufficient system response time to any adjustment before further adjustments are made.

3.11.2 Transition Operations

3.11.2.1 Transitioning from full operations to dormancy

Reduced power levels during dormancy may require reconfiguring the pump flow rates for the coolant loops. Possible reconfiguration or isolation of portions of the loops might be performed, which might save on overall pump power, preclude leakage, etc.

In the passive thermal system, internal heaters (e.g., walls) and external heaters (if applicable) will need to be set or powered during dormant periods.

Of importance is activating the monitoring and control software for the thermal system during dormancy. Automated functions or adjustments may be needed during the dormant period like pump flow rate changes or heater power adjustments.

3.11.2.2 Transitioning from dormancy to full operations

Activation of the vehicle systems previously dormant will require increased cooling capability. Likely the internal flow system will be increased first, and then the external flow loop will be ramped up. Reconfigurations may also be needed (e.g., reducing heater power, re-activation of isolated portions of the flow loop, radiators).

In order to prepare for full operations, any makeup coolant, mixture adjustments and/or purge of the flow lines should be completed, the flow system should be at nominal flow rates, and demonstrated or measured nominal temperatures should be achieved along with running on the software and algorithms for normal operations. During full operations, the monitoring and control system should automatically make needed adjustments to maintain the range of conditions required.

3.11.3 Contingency Operations

Contingency operations of the TCS likely would include adjustment of the fluid volume, flow rates and flow paths of the internal and external loops. Isolation of sections of the system may also be required (e.g., for leakage). Depending on the configuration during dormancy, active radiator area might also need adjustment (e.g., un-stow or re-activation of radiator sections or panels). In the most extreme cases, adjustment of power loads or vehicle maneuvers might be required to preferentially assist in heating or cooling portions of the vehicle.

3.11.3.1 Contingency Scenarios

Prior to habitation, the thermal environment must be established to support both equipment and crew. Similar to ECLSS, if the thermal environment cannot be recovered, or verified, prior to transfer from the

Mars Ascent Vehicle, then the crew must rely on some sort of portable equipment to maintain their thermal conditions, and may need to enter the habitat and manually recover or adjust the thermal systems.

This summary applies to the thermal system induced contingency events. Nominal functionality of the spacecraft thermal system is contingent on other systems' operations as well, and failures in these systems are not discussed here:

- Pressure Vessels: Excessive module leak rates, loss of critical resources from pressurized tanks
- Air/Ventilation: Loss of ECLSS air system
- Power: Loss of power
- Avionics: Loss of telemetry, loss of remote command capability, faulty commands to the LSS are not considered in this summary

3.11.3.2 Coolant Loop Failures or Off Nominal Conditions

Prior to habitation, the flow loops and overall system would have to be reconfigured to accommodate nominal power levels and operations. Failures or faults with the pumps, valves, fluid volume (e.g., leakage), etc., could preclude the system from being returned to normal operations. In that case, the crew would have to assume the vehicle has a non-habitable environment and don emergency equipment for troubleshooting and recovery activities manually or robotic repair would be necessary. Note that extreme deviation from the operating thermal range may require re-positioning or maneuvers of the space vehicle.

3.11.2.3 Monitoring Faults

The most critical functions during dormancy and prior to habitation are monitoring instruments or parameters, and automated analyses to implement control functions or adjustments as needed. If the avionics, sensors, etc. fail, automated adjustments to the thermal system would not be possible. In that case, the crew would have to assume the vehicle has a non-habitable environment and don emergency equipment for troubleshooting and recovery activities manually, or robotic repair would be necessary.

3.11.2.4 Major Radiator or Component Damage

Should the radiator section or panels or other major components (e.g., heat exchanger) be significantly damaged (i.e., beyond repair), replacement components must be installed or the flow system re-routed or isolated to recover as much of the active thermal system as possible. In the case of major leakage where insufficient makeup fluid is available, loss of the vehicle is possible. In addition, loss of major portions of the coolant loop system can result in loss of the vehicle.

3.11.4 Preventative Maintenance and Logistics Support

The spacecraft TCS is proposed to be designed and implemented with a "spare-in-place" philosophy, but may still require some maintenance and repair or spares replenishment during crewed operations. Components that are known to fail to start after long periods of inactivity, such as pumps, will be periodically cycled and/or used in alternating run modes or periods. Spacecraft system temperatures will require monitoring in order to control heat rejection turndown. Thermal transport loop fluid quantities will be monitored for leakage evaluation, and thermal transport loop pump performance will be monitored along with thermal transport loop fluid chemistry for corrosion evaluation. Preventative maintenance capabilities during dormant periods may include dosing of the thermal transport loop for correct chemistry maintenance. Corrective maintenance may include the need for remote valve control for radiator loop isolation to compensate for potential leaks. The capability to swap transport loop pumps through an automated transition may also be needed.

3.11.5 Autonomous Functions Needed

TCS components may suffer significant and catastrophic failures (e.g., pump failure), and the cabin environment and other equipment may require response faster than communication bandwidth would allow if initiated by ground controllers. The state-of-the-art is to have warning/flags and an automatic or commanded reconfigurations of the system, but future missions will require totally automated, “smart” monitoring and commanding, sometimes with “smart” analyses to determine what components or part of the system is faulty and what the best course of actions is. Technology gaps are identifying failures and the need to reconfigure, as well as the algorithms to analyze and command needed reconfigurations. The difficulty lies in the need for full automation under a wide range of system failures. There will be numerous sensors and parameters to be monitored and automatically evaluated (e.g., temperatures, pressures). Due to the dynamic nature of this system, there also should be consideration made to allow sufficient system response time to any adjustment before further adjustments are made.

3.11.6 Development Plan

The ATCS consists of many sensors and parameters to monitor and adjustments that could be made. As one of the critical systems on the spacecraft, the TCS directly affects the function of the rest of the spacecraft. The autonomy development for this system has two main tracks, the first being a systems engineering approach to designing the autonomous system, and the second an algorithmic approach to monitoring and controlling the system.

3.11.6.1 Systems Engineering for Thermal Control System Autonomy

The design of the overall spacecraft to reduce the complexity of the TCS will positively impact the development of algorithms to control and monitor the system. In particular, number and types of sensors could be reduced or configured in such a way to alleviate the cost in mass, power, and/or processing load by designing and selecting sensors that are optimized for the tasks. The configuration of the sensor network should be designed in such a way to aid in the development of the autonomous monitoring and fault detection algorithms to reduce the complexity of the overall system and improve system robustness. Sensor network design is treated more completely in Section 3.13.1. Likewise, the overall system design could reduce the number of faults possible in the system, as well as the number of configurations and possible actions that could be taken to command the system. These design considerations should be carefully considered alongside any contingency management and redundancy strategy. However, if it is possible to reduce the complexity of the systems, the complexity of the autonomous algorithms could also be reduced, which will overall reduce the failure modes and risk of the system, as well as the technology developments needed for the system.

3.11.6.2 Autonomous Algorithms for Thermal System Monitoring and Control

Several technology developments are needed to bridge the autonomy gaps that exist with human spacecraft thermal systems today. These technologies are all based around creating an autonomous fault detection, isolation, and recovery system for the ATCS.

3.11.6.2.1 Fault Detection and System Health Management

Reacting to or recording anomalies requires an accurate determination of the fault or failure. Detection of faults, either via using a fault model or without, is an important part of filling the autonomy gaps. While this technology is required for many subsystems (and is described in Section 3.13.2), these functions will likely need to be able to execute in a somewhat distributed manner, with subsystems making decisions as they are able to before a centralized system gets involved. Development on effective ways to deploy these distributed controllers is needed.

3.11.6.2.2 Data Analysis for Dynamical Systems

Though this technology applies to more than just the ATCS, the act of determining causation from a set of data from a sensor array is a difficult task, given the complexity of the system, the amount of data returning, the unknown environment, and the relative lack of situational awareness. The collection and processing of this data, along with other types of data that may constitute supporting evidence, requires significant processing capabilities and computing techniques today. The spacecraft must be able to locally draw conclusions from large but possibly incomplete sets of data using the processing capabilities resident on board. These conclusions may involve solving complex mathematical problems and applying detailed models. NASA investment in finding in situ data analysis solutions that can accommodate limited resources is important.

3.11.6.2.3 Planning

Because the control and configuration of active thermal system affects the spacecraft so profoundly, the ability to determine control actions to reconfigure and optimize the system is a complex and critical task. Constrained task planning that involves solving complex dynamical systems to find a provably correct and optimal plan is required to bridge this autonomy gap. This technology is discussed in Section 3.13.3.

3.11.7 Criticality and Redundancy

The TCS is a necessary and critical system for the space vehicle to operate throughout the mission. Sufficient redundancy is required to protect the crew and equipment, and enable mission success. The TCS has failure modes that involve immediate loss of function, but it can also suffer long term degradation. Redundancy in the TCS can be provided in different ways, such as having dual coolant loops, duplicating critical components and/or oversizing their capacity (e.g., pumps) by design, sparing in place or having spare components for replacement. As more autonomy is designed into the space vehicles, ensuring that the TCS is robust is critical and extensive ground testing in simulated environments to ensure the functionality, including in off-nominal conditions, becomes a priority.

3.12 Vehicle Systems Management

The vehicle systems management (VSM) subsystem provides coordinated oversight over the nominal, transition, contingency and maintenance operations of the spacecraft. The intent behind the VSM is to carry out some of the functions that the ground controllers would otherwise do when latency is less and bandwidth is more. Figure 9 illustrates the functional concept of the VSM. While the ISS's "Timeliner" functionality has some limited abilities that may approach what is described here, even that provides a weak state-of-the-art comparison to what is envisioned to be needed for this design reference mission. As such, there is no corresponding autonomy drivers section for the VSM- the drivers instead derive from the drivers in the subsystems that contribute to the integrated solutions handled by the VSM.

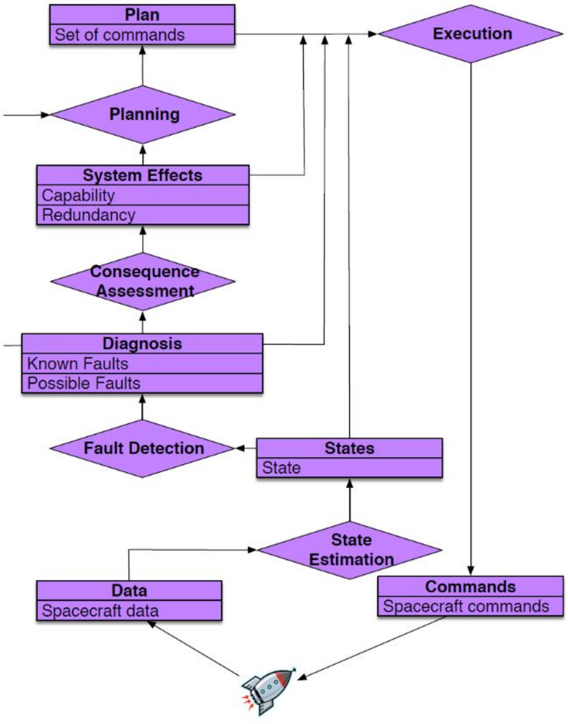


Figure 9: Diagram of VSM Functions

3.12.1 Nominal Operations

In nominal operations, the role of the VSM is largely to maintain and monitor the spacecraft. Though the locus of control largely will remain with the subsystems in most nominal modes, there are scenarios that require the VSM to take control. An example is the execution of procedures that are needed to maintain the integrated spacecraft state. The VSM will control the mission timeline, which will likely be uplinked to the spacecraft from the ground, and span days (perhaps weeks) at a time. The VSM will be responsible for ensuring that the tasks in the procedures are accomplished, and that any discrepancies are addressed appropriately. This will include planning or replanning as needed. The VSM will be responsible for the collection, organization, and prioritization of the downlink of data. This data will be used for the situational awareness of the system, both by the VSM itself and by the ground controllers supporting the spacecraft.

Data storage and downlink is an important autonomous function due to the limited bandwidth available. So, the VSM will need to be able parse through the data that is collected to ensure that only the essential data is sent down to ground controllers. Like the other functions that the VSM is responsible for, data processing must be conducted on an integrated spacecraft level to ensure that the optimal set of data is downlinked. This vehicle-wide data processing is also necessary for fault detection and response.

3.12.2 Transition Operations

For both the transition into dormancy and the transition from dormancy to crewed operations, the VSM will be responsible for the execution of many of the procedures to bring systems down or up, as required. For example, many of the ECLSSs will continue to run for some period after crew leaves in order put the spacecraft in the proper state. While ground controllers may be involved in these procedures, the VSM will be essential to procedure execution due to the latency and communications bandwidth.   The VSM's role will include taking appropriate corrective actions if and when problems are detected and identified by the VSM. Transition operations may also require autonomous planning and replanning actions in order to ensure that any anomalies or unexpected events during the transitions will not put the spacecraft into a critical state.

3.12.3 Contingency Operations

The VSM will owe most of its complexity to the functions needed during spacecraft contingency operations. Three broad types of contingencies will be described during this section. First, the VSM will lead the detection, isolation, and recovery for any emergencies. The VSM may also perform detection, isolation, and recovery if any subsystem component failure has effects reaching beyond the individual subsystem. And finally, the VSM will be responsible for analyzing and determining responses due to model inaccuracies or unexpected events caused by the environment.

3.12.3.1 Emergency Management

Because spacecraft emergencies typically impact every spacecraft subsystem, the VSM will be active in the detection, identification and localization of fires, leaks, or other emergency conditions. The VSM will have access to all the sensor networks and robotic agents, and will be able consolidate the data and resources needed to respond quickly to isolate and recover from the emergency. Recovery will likely require reactions from several control systems. For example, detecting and responding to a leak may need ECLSS, GNC, and propulsion to react, among others. Robotics may be required to repair or clean up from the emergency recovery, and the VSM would direct these assets appropriately.

3.12.3.2 Component Failures

The VSM is responsible for aiding in fault detection when subsystems need assistance to do so. Given the integrated perspective that the VSM has, extra information can be used to diagnose problems. If the problem affects other subsystems, or solving or containing the problem has consequences on other systems, reconfiguration of the overall system to accommodate or recover from the fault will be the responsibility of the VSM. In order to determine the appropriate locus of control, the VSM will need insight into subsystems' local FDIR processes; the VSM can use component configurations, local fault detection conclusions, and component data to obtain control and manage the system as a whole. Alternatively, a subsystem controller can request help from the VSM if it cannot determine the best course of action on its own.

3.12.3.3 Unexpected Events and Model Inaccuracies

Unexpected events are distinguished from faults and emergencies. These events are assumed to be environmental in nature, e.g. space debris instead of a power system fault. Unexpected events may require detection and recovery; space debris may trigger a maneuver, radiation events may cause shutdown of avionics and software, and so on. The VSM will take responsibility for detection and response to such events.

The VSM will be responsible for identifying and correcting system level errors that result from any number of situations that will be covered under the title of model inaccuracies. These models include configuration of software in either subsystems or the VSM itself, environmental effects over time on subsystems or the spacecraft, or expected degradation of components. Since the VSM will have access to all of the data, the analysis of model performance will be the responsibility of the VSM. The time criticality of the problem will dictate how much intelligence, planning, and reasoning ability is required for the VSM, as problems with long time horizon will require mostly detection, data collection, and correct packaging of data to send to ground controllers.

3.12.4 Preventative Maintenance and Logistics

The VSM will be responsible for executing any maintenance procedures needed for maintenance of the dormant spacecraft. Some subsystem components that are generally off during most of the dormancy phase may need to be cycled or turned on periodically to ensure successful transition back to full operations for the journey back to Earth. The VSM will be responsible scheduling and executing the cycling operations. In certain cases, overriding the subsystem control system may be required in order to do the maintenance tasks (i.e., the jet tables in the propulsion subsystem). If robotic maintenance or logistics tasks are required, the VSM will be responsible for directing all agents (subsystems and robotics) to complete the task based on the skills and abilities that are required for the task and that can be provided by the particular agents.

3.12.5 Autonomous Functions Needed

In its simplest form, the VSM implements the ‘sense-think-act’ loop, which is one way of characterizing the basic building blocks of autonomous systems. The following elaborates on these building blocks to identify more traditional mission operations functions:

- Monitoring. This function includes the collecting of information from spacecraft subsystems, and the processing of that information to assess the state of the spacecraft and the spacecraft external environment. This function includes trend analysis, anomaly detection, and prognostics.

- Planning. This function includes ensuring that near-term future activities for all spacecraft systems are known and do not violate any constraints. For human spaceflight systems, the Earth-based mission control centers plan 2-3 weeks into the future; the scope of an autonomous system is open to debate. This function includes sequencing system automation scripts and procedures, and building extended tactical plans for each subsystem comprising collections of scripts.

- Executing. The execution function enacts plans. This includes determining when to invoke or terminate scripts, and invoking and terminating those scripts at the appropriate time or under the appropriate conditions. While execution may occasionally require sending individual commands, for the most part this function will operate at higher levels of abstraction.

- Fault management. This function includes fault detection (identifying the presence of a fault) and fault isolation (identifying the cause of a fault). Note that fault detection and isolation are considered here as spacecraft- or system-wide functions, and will take as input subsystem and component level fault detection, isolation, and recovery (FDIR) output and fault codes.

For a vehicle of the size and complexity as the DSG or Deep Space Transport, the VSM may in general consist of multiple interacting monitoring, planning, execution and fault management modules. Traditional flight control is divided along system lines (e.g. power and ECLS), and this forms a natural architectural breakdown for the VSM components. The individual components must, however, be integrated. Plans for power and ECLS must be integrated if they are built or managed separately; fault management functions may similarly be partitioned and subsequent results integrated. The execution function closes the loop on the monitoring, planning, and fault management functions; this provides the capability to respond to unexpected events, perform active fault diagnosis, and fault recovery.

The VSM must have a notion of its 'authority' or 'responsibility.' In some mission phases or contexts, the VSM may be permitted to take action on its own; in others, the VSM may need to delegate responsibility to the control center. Notably, lower level software functions may take responsibility, meaning that the 'locus of control' will generally shift between elements of the onboard system, as well as between the onboard system and the mission control center. The traditional means of assessing these conditions is to assess the time to criticality of faults or unexpected events. This must be coupled with the degree of certainty that the VSM can take the appropriate response.

Like other software systems, the VSM should be designed for change. This is especially important since operational constraints will change significantly over the lifetime of the spacecraft. All of the key VSM components can be built as generic computational engines that accept as input a run-time configuration. These configurations include, but are not limited to:

- Operating constraints. This includes plan action descriptions, resource usage of actions, action duration, etc.

- Existing scripts (nominal, fault detection and fault response) as well as primitives to generate new scripts (as a result of planning).

- FMEA/FMECA and fault signatures.

- Monitoring configuration. This includes limits, anomaly rule bases, prognostics rule bases.

- Interface to lower level system(s) (especially if they change). Incudes FDIR and other state information from system-specific software functions, as well as sensor data.

3.12.6 Development Plan

The VSM development plan is divided into roughly three parts. 1) VSM component-specific tasks, 2) VSM architecture and integration, and 3) System Engineering, Verification, Validation and Certification. Each of these areas includes specific work items for the major VSM functions (monitoring, planning, execution and fault management).

3.12.6.1 Component Specific Development Tasks

The basic software development task for each VSM component is the selection of the best algorithm, or algorithms, for the reasoning engines.

A combination of the characteristics of the system, and the underlying mathematics of the problem or problems to be solved, will drive selection of suitable algorithms. For instance, robot motion planning is fundamentally a matter of kinematics, dynamics, and continuous mathematics; planning algorithms for managing power systems may include some continuous math (reasoning about currents, voltages, and batteries) but at the more abstract level require discrete selection of power modes (including unpowering systems). Similar considerations apply to fault management. Considerations for planning and plan execution include whether to explicitly reason about uncertainty (e.g. reason about the probabilities of unknown action outcomes), or whether to revise plans when new information is received.

Resource usage (central processing unit, memory, runtime) will also drive algorithm selection. Some of the VSM functions (planning, fault management) are known to be time intensive. Some (monitoring, anomaly detection, prognostics) can be memory intensive. Generally, time and memory can be traded with generality of the problem to be solved; work is needed to identify 'nice' versions of the computational problem whose solutions are acceptable for operational use. Additionally, while some algorithms have been ported to embedded processors to characterize resource usage, considerable work will be needed to move more and more novel algorithms to embedded environments to characterize resource usage, and inform algorithm selection for the large-scale problems that are expected for these spacecraft.

A common way of managing the complexity of planning, in particular, is to restrict the scope of future actions, also referred to as the planning horizon. Alternatively, planning can be performed at a higher level of detail in the short term, with more abstract plans for further into the future, where there is less certainty. These considerations will vary system to system.

There are generally different approaches to solving the same problems. This means tradeoffs need to be made, and criteria for evaluating trades is key. Are the problems large enough that algorithm run-time is the pacing item (which can be the case in planning or fault management)? Are existing algorithms simply not solving the full problem (e.g. applying a discrete technique to what is fundamentally a continuous problem)? Is it time-critical to respond to a problem or unexpected event, or is it more important to find the best possible solution even if longer computation time is needed? Ensuring that the metrics that drive tradeoffs are known, and then performing the tradeoffs, will be key to fielding the right algorithms for each VSM component.

3.12.6.2 VSM Architecture, Integration, and Locus of Control

This discussion is focused on the architecture and integration of VSM capabilities. The integration of VSM capabilities with other software systems will be covered in the next section.

When a function is decomposed across systems, integrating the results will be required. Exactly how functions are decomposed is key: are functions divided up in a 'peer-to-peer' or 'divide and conquer' approach, or are they decomposed in a more 'hierarchical' manner? In some cases the decision of how to divide the labor may be natural. For example, scripts are plan building blocks, so planning can be divided up hierarchically into 'automatic scripting' and 'sequencing of scripts.' In other cases, e.g. power and ECLS planning, the relationship may be more of a peer-to-peer, with synchronization and integration. The same considerations hold for fault management. The situation becomes even more interesting when considering different fault management algorithms with different strengths and weaknesses; a discrete, test-based approach can be combined with a hybrid model that integrates both discrete and continuous reasoning. The different approaches for decomposition and integration must be explored to identify the most suitable integration strategies.

The hierarchical versus peer-to-peer discussion above is related to the broader issue of locus of control and authority design for the VSM. First and foremost, the considerations for locus of control must be understood. Some of these will arise directly from requirements, which in turn will be driven by time to criticality considerations, especially for faults. Others, however, will need design and development. As noted above, the VSM must be cognizant of other cases in which onboard systems will take control. These can be lower level controllers for systems (e.g. power system or robots). Locus of control within the VSM is also important if elements of the VSM are designed in a peer-to-peer fashion. Finally, there is the question of where and how to represent locus of control considerations. What information drives locus of control? Is locus of control implemented as part of the execution system? Are they part of some execution system configuration, or something else? An additional factor in locus of control is the confidence the VSM has in the decisions it makes. In planning, for instance, if two plans look equally good but have consequences that are hard to quantify, should the VSM simply pick one, or request clarification from ground? For fault management, if a fault cannot be uniquely identified and the fault responses have different consequences, the same consideration applies. Research is needed to decide how such cases are best handled.

Above and beyond these considerations is where VSM components run. If they are run in multiple processors, or even multiple threads on the same processor, then communication between components must be designed. This is important even if the VSM components are hierarchically organized.

Since the VSM will be making decisions and acting on them in the presence of unexpected events, the VSM will need to record those decisions that are made, and also explanations of these decisions. As an example, suppose a power system fault occurs, and the VSM opts to unpower a system to preserve the battery. The VSM must record not only that the load was unpowered, but why (e.g. to avoid violating a battery depth of discharge limit. Work is needed to identify the kinds of explanations that are needed, and how (or whether) to produce them (as opposed to reconstructing them on the ground from smaller amounts of data). These records will be sent to the mission control center. This latter point requires prioritization of downlink data; while this is common practice today, further work is needed.

3.12.6.3 System Engineering, Verification and Validation

The VSM design above imposes a variety of interface constraints on system-level software. Many of these requirements are described in the sections above, but to collect them all together:

- Command. The VSM must be able to command lower level software functions.

- Data. Lower level software functions must report data to the VSM.

- Locus of control. The VSM must be able to negotiate control with lower level functions.

- Transparency. Lower level function commands and state changes must be recorded and broadcast to the VSM.

Ensuring these requirements are met, on both sides, is part of the VSM system engineering task. There is clearly wide scope in how best to design these interfaces. Research is needed to explore this design space.

Generally speaking, V&V of autonomous systems is an open challenge. A discussion of this technology can be found in Section 3.13.4. On the one hand, the software components of the VSM must be verified and validated. Much of this technology is challenging to V&V and test; although techniques have been applied to deep space and planetary surface robotics, it may be difficult to scale to the size of the spacecraft. As mentioned above, the VSM (as with many other software components) must be designed for change and be reconfigurable. The configuration of many of these systems, especially the planning and fault management systems, uses a variety of modeling languages. It is important to recognize that these models are not used merely for analyzing the VSM behavior, but are in fact part of the VSM software run-time configuration. As such, ensuring model correctness is part of the VSM V&V and certification task, and is another area of significant open work.

A related problem of model correctness is that of model composition. This problem arises for two reasons. The first is comparable to challenges in large software engineering projects; the fact that the models are built separately and must be integrated. This challenge will arise simply because of the size of the system[2]. The second problem is more difficult. Disparate technologies use different modeling paradigms. This makes the challenge of integrating models more difficult, because now the semantics of the models need to be matched. The problem is compounded because there are multiple languages and semantics for technology enabling the same capability (e.g. planning), as well as different semantics between capabilities (e.g. planning and fault management). Existing work in this area involves the use of ontologies and ontology languages (e.g. Web Ontology Language or OWL and OML/SysML), ontology reasoners (e.g. Pellet for OWL, Modelica for SysML), and custom mappings from these languages to those used by reasoners (e.g. SysML to TEAMS for fault management). Work is needed to continue refining these developments.

3.12.7 Criticality and Redundancy

The VSM is considered critical, in that it must be running at all times to ensure nominal execution of the mission timeline, and to ensure unexpected events or faults do not disrupt operations.   Essentially, the

[2] I have a running example I like to use with planning. Suppose I require a planning model to control a lightbulb. The bulb has a switch. The bulb is also on a power circuit. The circuit must be closed and the switch on for the light to be illuminated. If one person is modeling the lightbulb and switch, and another person models the circuit, and they don't agree on the semantics of 'states', (e.g. if one person thinks 'on' is '0' and another thinks 'on' is '1') then when the model is joined together, things are broken. This is a problem that exists **even when using the same modeling language.** When multiple languages are used, it's even worse.

VSM is in the same category as the rest of the flight software. This means the VSM can be disrupted or disabled by the same problems that can impact flight software: radiation induced hardware faults such as SEUs or hard computer faults, loss of power to computers, and software defects (either executable or configuration errors).

Redundancy for the VSM is accomplished by replicating VSM software, either as multiple processes within a single computer or across multiple computers. This will provide protection against many of the critical failures that can disrupt or disable the VSM. More complex protections could include using non-avionics computers in the shirtsleeve environment, in different parts of the spacecraft, or even in robots. These more complex protection mechanisms have significant implications on the avionics architecture, specifically on the data and command pathways to compute assets that typically are not used for these purposes.

3.13 Development Plan for Common Technologies

There are several technology developments that are needed across many subsystems. In the interest of efficiency, the common points of each of these technology development plans are described in this section. For the subsystems that would benefit from these technologies in any specific way, the individual development plan will list the specific points and refer back to these common sections.

3.13.1 Sensor Networks

Many subsystems rely on situational awareness via sensors. In many cases, networks of sensors will be needed in order to identify the source of a fault or failure. Sensors in these networks may be heterogeneous or homogeneous, may require power or not, but for all, these sensors will cost mass, volume, and data logging, storage, and processing. Sensors have mass, as do the wires needed for their power and/or communication, and for the processors needed to collect and reason about the data. Wireless sensors, and sensors that can be interrogated by RFID or other means, reduce wire weight, but lead to other system integration questions such as transmitter location and frequency interference. Many sensors need to be powered, but all sensors must be read by a device that requires power. Sensors also add complexity to the system, in that failure of sensors is possible, and so that must be accounted for in both the algorithms that use the sensors and in the risk analyses for the functions that the sensors enable. Failures in distributed sensing must be accommodated via the overall design, such as in the robustness of the sensors to failure, the distribution function of the sensors being designed to accommodate a certain percentage of failures without affecting overall performance, the ability to replace sensors via robotic means, or some combination of these things.

Therefore, investment should be dedicated to several facets of this problem. Developments are needed both for the design of robust sensors for spacecraft, given the environment and other considerations, and for the overall sensor network design to reduce the cost while taking into consideration its intended use. Developments in sensor design to be able to use the same sensor for many different needs would reduce the overall footprint of the sensor network, though these may increase the complexity of the data logging, distribution, and analysis algorithms. Understanding the needs of each type of sensor is important for developing sparse monitoring techniques that can reduce the power and data processing footprint of the network during quiescent phases. The density and composition of sensor networks rely on the functions that need the data provided. For example, many systems require sensors to give the location of an event or a fault. Developments in algorithms that use the sensor networks to do these sorts of data analysis should be developed concurrently with the design of sensor network configuration and layout in order to optimize the overall system. Keeping the autonomous functionalities in the trade study when considering these systems will give the overall system a better chance of success.

Finally, certain types of sensing exist on board spacecraft presently due to the mobile sensor packages naturally built into every crew member. As such, a trade should be undertaken to understand the costs and benefits of mobile robotic sensor packages on the needed situational awareness for the functions using the data. These mobile sensors would likely reduce the overall cost of the sensor network by requiring fewer sensors, wires, and potentially less complexity in the system. Sensors on a mobile robotic platform could be larger, require more power, and produce more data than sensors in a distributed network, which may be beneficial for some applications.

3.13.2 System Health Management

Understanding the state of the system is something that is required for each subsystem and the overall spacecraft. Many refer to parts of this as fault detection, fault identification, or other fault determination terminology, but the concept is a bit broader than that. System health management involves not only understanding if faults have occurred, but also understanding whether the operation is degraded at all, and if so, why. The problems are complex and can involve discrete or continuous spaces to reason over, but in each case, the goal is the same: to understand the state of the system.

Detection of faults can occur via using a fault model or, in certain cases, without, and in a distributed or centralized manner. Because this technology is required for many subsystems and reactions may need to occur at different time scales, fault detection and isolation can be implemented using a ‘divide and conquer’ approach, with subsystems making decisions as able before reporting to a centralized system. At the same time, many subsystems may use similar, or identical, fault management technology, but synthesizing the results of multiple fault management engines is an open topic.   Development on effective ways to deploy these distributed models is needed.

The primary driver for NASA investment in these technologies are the unknown environments and conditions into which the spacecraft will be deployed. Without the ability to model the systems and the environment fully and likewise without the ability to train the system using relevant data prior to launch, technology advances will need to be made to allow the system health monitor to either adjust in situ as “nominal” changes due to new knowledge and exploration, or to log and work around likely model deficiencies prior to reconfiguration from ground.

3.13.3 Planning

Because the control and configuration of many of the subsystems affect the spacecraft profoundly, the ability to determine control actions to reconfigure and optimize the overall system is a complex and critical task. Automated planning that involves solving complex task ordering and resource allocation problems is required to bridge this autonomy gap.

Automated planning includes a variety of technical disciplines. For instance, kinematics and dynamics feature prominently in robot arm planning, but are not required for power systems reconfiguration planning. Solving complex dynamics constraints may be needed in some cases for thermal and ECLS systems planning, but may not be required if the problem is suitably abstracted. Planning is a computationally hard problem, but a variety of sophisticated techniques can be used to provide good quality plans in a timely manner. Work is needed to identify the best algorithms to solve each type of problem; tradeoffs between plan quality, time required to plan, and computational resource constraints must be performed.

Automated planning will be needed for several different spacecraft subsystems during dormancy. Primary among these are the power, thermal and ECLS systems, but planning may also be needed for avionics and robotics systems, both external and internal. Automated planning will be needed only seldom during nominal operations; monitoring the nominal plan and managing unexpected events should lead to few disruptions. When a fault occurs, automated planning will be needed to refine fault identification and implement fault recovery. As with fault management, planning may be performed for individual subsystems in a ‘divide and conquer’ mode, but integration of plans for each subsystem will be generally required. Research is needed to determine the best strategy to build and integrate plans for each separate system.

While this technology is likely to apply to systems on Earth as well, NASA investment is required due to the very low TRL of this capability.

3.13.4 Verification and Validation of Autonomous Systems

Because the functions that remain active during dormancy are critical, it is essential that any autonomous functionality be sufficiently tested and guaranteed before deployment. The assurances required of these autonomous functionalities to gain the trust of ground controllers constitute the V&V of these systems. The challenge becomes that the system will be deployed in largely unknown and unanticipated conditions. As such, the ability to fully test the system, particularly systems that employ learning technologies or model adaptations, do not exist within the methods available currently. New ways of testing and monitoring systems for the satisfaction of their intended functionality, both prior to launch and during deployment, are needed. NASA investment is essential to drive the advancement of these technologies due to the unique requirements that exploration imposes on the system design life cycle.

Another challenge is that the verifications must be given, in many cases, in a nearly absolute sense, such as declarations that the system can “never” fail. The testing and monitoring methods that will be required to satisfy the necessary assurances while accommodating the new types of systems have not been invented yet. Provably correct constructions, assume-guarantee contracts, runtime monitoring, and some formal method approaches are all candidates towards feasible solutions to the V&V problem. NASA investment in this area is essential to gaining the trust in these autonomous systems that are necessary for missions such as this design reference mission to Mars.

4.0 Recommendations

The detailed analysis in Section 3.0 was carried out with the intent to identify technology gaps by first defining the concrete reasons why further development of autonomous systems is necessary from a practical operational standpoint, and then by defining the functions that will need autonomous systems to cover them during the uncrewed period of this design reference mission. The autonomy drivers serve as the impetus, the necessity for autonomous systems development. Each system analyzed has an identified gap; no system is currently operated on human spacecraft as it would need to be operated for the communication constraints imposed by this mission. In addition to this, every system has functions that are needed for the various stages of operations that are associated with the uncrewed phase that are currently covered by crew or ground support currently. While the uncrewed phase has been commonly referred to as dormancy, no critical subsystem will have the luxury of performing no actions during this time.

As such, there is an incredible amount of information contained in the analyses above. This section will attempt to summarize while providing insight into lessons learned while conducting this study. The recommendations are meant to inform mission and spacecraft designers about the needs that must be considered when incorporating the necessary autonomous systems technology. They are also meant to inform technology and architectural investment strategies in the area of autonomous systems development.

While the scope of this design reference mission is necessarily limited, other mission phases and other mission types could benefit from some of the discussion of the recommendations that are provided below. Within this mission, autonomy for ground controllers and autonomy for crewed mission phases are out of scope but some details are provided in Appendix B and Appendix A, respectively. Autonomy needs for surface habitats are not covered at all, but much of the information provided pertains to this mission phase as well.

4.1 Systems Engineering Enables Autonomy

The successful integration of autonomous capabilities into a spacecraft necessarily start with a solid approach to systems engineering. Because the ability to operate independently from ground (or crew) control requires knowledge and control of the entire spacecraft, an integrated system design is essential to the success and independence of the overall system. Practically, this means that system designers should concentrate on several things during spacecraft development. First, design choices that result in a reduction in system complexity will enable more robust reasoning about the system. Systems that are less complex need less complex algorithms to reason about them, and as a side benefit, will also allow more human visibility into system reactions. Along the same vein, reductions in dependencies between subsystems enables more local, componentized decision making. This will also enable less complexity in autonomous solutions due to the inherent reduction in the overall state space or fault tree that must be considered. **The single most important consideration in system design is understanding the interconnections and interdependencies between the spacecraft subsystems.**

From a fault response standpoint, system engineering consideration is important to attempt to increase the time to criticality of all possible faults. Similarly, careful decisions in fault trees and redundancy increases TTC by requiring more than one failure before decisions become critical. Longer reaction times enabled by good design allows more time for computation and review. Longer time to criticality also provides more options to the autonomous functions and ground controllers. Spacecraft that are naturally more robust to failure are easier to operate and therefore more inherently autonomous.

Because the spacecraft will be a complex integrated system, even if system design focuses on simplicity and independence, subsystem connections will happen and system design should acknowledge this. Interfaces, fault trees, and autonomy should be designed to accommodate these connections. Subsystem design cannot happen in isolation, and system engineers that focus heavily on designing the interface requirements (as well as the testing, verification, and validation requirements) must be integrated with each subsystem design team as well as on a team of system engineers. Strict compliance to interfaces, configuration management, and other best practices will be more important than ever to ensure successful spacecraft autonomy.

In addition to the careful design of subsystem interfaces, interfaces to autonomous functions must be carefully considered. The authority to act (locus of control) and delegation of functions is important, particularly between subsystems (distributed intelligence) and the Vehicle Systems Manager (centralized intelligence). This will provide a dictionary of tasks and functions the autonomy can do, and which system is authorized to accomplish these things at any given time. One can imagine the need to reconfigure the distribution of control authority between mission phases, such as crewed versus uncrewed or even near Earth versus Martian orbit. This distribution of control is important for identifying the exact scope of the autonomous functions needed in a subsystem, in robotic agents, or in the VSM, which will be important for understanding power and processing needs for each of these things. This document provides a start along this path, but careful systems engineering is needed to further develop this topic.

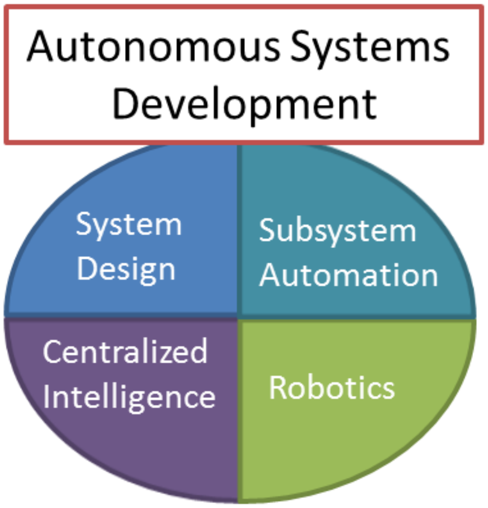


Figure 10: Developments contributing to spacecraft autonomy

Communication flow up, down, across subsystems and the VSM is essential to ensure proper data visibility and crisp interfaces. Careful design of data flow, storage, and processing is needed to achieve the optimal collection and delivery of data in this constrained environment. These interfaces are as much dependent on software design as they are on hardware. As such, *hardware and software integration must start at the beginning of any design effort*. Integration with ground control is also a very important consideration. Autonomous control should strive to be explicable at all times. Ground control or crew members should never have to ask why an autonomous function did what it did. This requires careful design of the system for human interaction, and puts constraints on data and communications designs.

Autonomy will be achieved using several contributing factors, shown in Figure 10. In some cases, system design improvements will go a long way (i.e., fire suppression). Automated responses are currently lacking in a few places, where subsystems could handle local faults autonomously (local subsystem automation). More integrated autonomy software, diagnostics, prognosis, and other higher level functions are needed, as discussed in the VSM analysis section (centralized intelligence). Finally, manipulation, mobility and inspection have roles in contingencies and preventative maintenance (robotics).

Fitting with the recommendations on system design are two more topics. First, several trade studies are recommended to be conducted in very initial stages of system architecture design, as these decisions will drive significant investment decisions. Second, many architectural requirements are driven by the autonomy needs outline in this report. Examples of these requirements are discussed in a subsection that follows.

4.1.1 Trade Studies Required

In the process of conducting the dormancy autonomy analysis, several high-level architectural choices became apparent. These choices generally involve trades on mass, cost, complexity, and risk postures, but they have significant effects on the assumptions that system designers would make when generating system requirements. They also have significant effects on the research and technology development efforts that receive should investments from NASA. As such, these trade studies are recommended to be completed immediately for the best chance of future mission success.

All of these trade studies essentially come down to defining the role of robotics in maintaining the spacecraft in uncrewed periods. While it is expected that robotics will play some role, it is very unclear how much of a role makes sense, given the current state of the technology, risk, and the departure from current operational paradigms. However, because of mass, power, cost, complexity, processing, and data storage constraints on board the spacecraft, it is very important to conduct trades with due diligence, as it is expected in most cases, solving manipulation, mobility, and inspection problems without robots will have a higher cost in several of the aforementioned constrained areas.

The systems architecture trades needed are as follows. First, a trade needs to be conducted on sensor arrays versus mobile sensors for the many autonomous functions that require distributed sensors in order to provide the situational awareness needed for control and human understanding. The trade should be careful to consider all functions that will require sensing, both inside and outside the spacecraft. The types of measurements being taken should be compiled, and any cross-over between systems should be identified. Sensor density should be considered based on the functions required of the data collected (i.e., localization of a leak within areas of square centimeters will have different needs than understanding the air temperature in a module), as well as the robustness of the sensors (expected time to failure), and the spares strategy employed (in place redundancy versus robotic replacement or repair- detailed below). Power needs, cost, processing, data storage, and bandwidth requirements should also be considered. These things should be weighed against the needs that go along with robotic inspection systems, and a consideration of what types of sensors and measurements would be appropriate for a mobile platform. The solutions being pursued in other types of industries (oil and gas, nuclear power plants, bomb disposal, etc.) should be considered as well.

The next trade that needs to be studied is self-actuated valves, switches, and other interfaces versus mobile manipulation. This is very similar to the previous trade in that places where self-actuation is needed across all subsystems must be noted and certain data must be collected. Like the sensors, self-actuation requires mass, complexity, processing, data, sensors, and power. Robotic mobile manipulators could be used in some or many cases instead, but this also costs mass, power, data, processing, sensors, and complexity. The other trades should also be considered, as the same robotic platform may be able to fulfill multiple roles, which will reduce the overall cost of the robotic system. Robotic mobile manipulation also has significant technology gaps, which increases the risk of this solution. Analysis of the state of robotics technology required to create a solution with the proper robustness given the criticality of the tasks the robot may be asked to do should be a component of this trade study as well.

The final system-level trade considered here is analyzing in-place redundancy and component robustness versus a “replace and repair” spares philosophy. Investments for the in-place option will require consideration of mass, power, and complexity in addition to the overall cost of components, which will be higher. The investments in the “R&R” option will be centered upon the robotic agents needed to conduct this. A discussion on overall risk posture of the mission is also appropriate for this study, as the attitude towards failed components that are part of a redundant solution to critical system risk will greatly influence the requirements for this trade. The same cost functions will be present as have been listed for the previous two trades. Like the previous trade, the analysis on the cost and complexity of the robotic solution should be considered with other trades on the subject. If the same robotic agent could fill multiple roles, the relative cost is decreased for each function it will perform.

There are other architectural decisions that likely merit trade studies, such as propulsion system refueling, logistics arrival and management during uncrewed phases, Martian satellite systems for relay communications, among others. While these decisions affect the architecture and autonomous technology development for the spacecraft, their effects are more localized and so are not treated in this section.

4.1.2 Architectural Requirements

This design reference mission has generated several architectural requirements for the future exploration spacecraft that will experience uncrewed periods. These center around the seamless integration of autonomous capabilities that will allow the spacecraft to handle operations and events that have times to criticality that are shorter than the communications constraints will allow. These requirements cover overall system design as well as some subsystem design requirements. A separate grouping of requirements specifically reflects some of the data requirements and considerations that result from autonomy needs for dormancy operations.

4.1.2.1 System Design Requirements

The spacecraft shall utilize highly reliable radiation tolerant components optimized for use in deep space.

Rationale: Failure rates need to be reduced to the extent that both necessary redundancy and reduced sparing can be met for mission needs.

Critical subsystem design shall incorporate an initial failure of fail-operational.

Rationale: Critical systems should not go directly to fail-safe with a single failure.

The critical spacecraft systems, upon a failure that is addressed through redundancy, shall remain on the backup string until which time either the ground can intervene or analysis indicates that the ground cannot intervene in timely fashion, in which case autonomy will intervene.

Rationale: Onboard autonomy, where it is not necessary, adds to onboard complexity.

The spacecraft subsystem critical components should be designed to be both Intravehicular Robotic Activity and Extravehicular Robotic Activity compatible.

Rationale: Potential dormancy periods are of sufficient duration that IVA/EVA R&R may not be practical. Robotics enable a "common spare" approach to replacing and/or repairing avionics boxes, pumps, valves, sensors, etc. Without robotics, critical items must have in place redundancy to ensure failures are accommodated. Overall this is likely to increase the mass of the spacecraft, which must carry around all failed components for the full dormant period instead of allowing for a small number of spares to cover common hardware (i.e., valves, avionics boxes, etc.). Using the "common spare" approach does impose other requirements on the system, such as accessibility for the robotic system to do the repair or part replacement.

The spacecraft system shall consider emergency identification, response, and recovery stages as an integrated part of the system design

Rationale: Robotics can serve as a human replacement for emergency response procedures. For example, a robot can clean up after a fire suppression system has been deployed. A robot could identify precise locations of leaks or potential fires. A robot could repair a hull breach or other mechanical failure to preserve resources. If robots are not used for these things, either the system must be designed in a way to handle these emergency events or the risk posture of the overall mission must be ready to accommodate recovery from emergency as part of crew ingress to the return spacecraft.

Autonomous capabilities and control authority shall be distributed amongst the subsystems.

Rationale: This allows the allocation of levels of intelligence or capabilities to improve processing behavior, performance, and redundancy. This is an architectural departure in some respects, however there are aspects of this approach being used on Orion, particularly in ECLSS, where devices are able to operate automatically if the central processor is down or disconnected.

4.1.2.2 Data-Driven Requirements

Data interrelationships and interdependencies shall be established on the spacecraft level and on the subsystem levels.

Rationale: Data-driven design will provide clear interfaces for the distribution of authority and the locus of control for the autonomous system. This will ensure coverage of autonomous response without overlap.

The spacecraft system shall provide necessary data to support a centralized intelligent system in support of autonomy operations.

Rationale: Adjustments will need to be made during long-duration dormancy periods as to what is being recorded, what may have failed, and what state changes may have occurred in support of onboard autonomy. This data requirement pushes the need for data-driven design happening early in the spacecraft design process. The data-centric approach would drive out the software concept for operations, software requirements and subsequent software system design. As stated prior, data interrelationships determine how the software system should be designed in relation to VSM and the subsystems.

Alternate data sources shall be identified to mitigate incomplete or corrupt data within or between subsystems.

Rationale: Sensors will be essential to data collection on board the spacecraft. Redundancies in sensor arrays must be established in order to ensure sufficient coverage in the presence of failures. Robotic sensors enable a reduced built-in sensor array. Situational awareness can be supplemented with mobile sensors, likely reducing the mass of the overall system. Heavier and more power-critical sensors can be deployed on a robotic platform and used when and where needed. Requirements imposed on the overall spacecraft by this approach can include installing navigation aids inside the spacecraft, allowing for robotic berthing on structure (such as handrails), providing for recharge docking stations for the robot, and requiring wireless data accessibility throughout the spacecraft.

4.1.2.3 Subsystem-Specific Requirements

The avionics system shall support what telemetry is to be stored and forwarded based upon resulting states from autonomy intervention.

Rationale: The loss of communication periods will be of such a duration that not all telemetry will be able to be stored easily for eventual retrieval. What telemetry will need to be downlinked will be driven by what decisions were made autonomously onboard so as to understand the current state from the previous state.

The avionics system shall have the ability to distribute FDIR where no component is completely reliant on solely diagnosing itself.

Rationale: A faulty avionics system component may be limited in its ability to do its own self-diagnosis.

Communication systems on the spacecraft and on the ground shall be designed to work together to autonomously recover connections as necessary.

Rationale: When there is a loss of communications onboard the space vehicle, as part of its FDIR, it drops to a default configuration. Likewise, when there is a configuration change in the communications system, *the ground station does not automatically detect nor change its state to the new configuration. The mission control center operators have to detect the loss of communication or determine the configuration change and have to request the ground station to change the settings to the new configuration. If the ground stations are upgraded to automatically and adaptively sense the changes in the received signal and reconfigure its settings, this would minimize the duration of the loss of communications as well as provide some autonomy in the ground station and control center operations.*

4.2 Autonomous Technology Investments

The analysis on the autonomy drivers, functions required, and autonomy gaps for each subsystem and for the overall VSM beget a list of technology investments that are needed in order to achieve the minimum functionality required for the design reference mission. However, for many of these technology areas, only a simple review of the state-of-the-art and determination of TRL was given. Investments into these areas should start with, if it is not already known, a detailed assessment of the state-of-the-art, a benchmarking activity with industry, including the expected development plan that the industries working in the area will take over the next several years, and a determination of the technology development roadmap for the area to achieve the appropriate readiness level by the time the mission needs dictate. This section will list some of the technologies that are common to many or all of the subsystems as these are broad areas of focus that will have the most impact on the autonomy of the overall spacecraft. It will also give some recommendations on investment priorities based on current knowledge of the state-of-the-art and technology development needs.

4.2.1 Common Technologies Needed

While these areas are covered more completely in Section 3.13, they are reiterated in this section as a recommendation of broad investment strategies for NASA technology development programs. These are recommended because they will have the farthest reaching impacts in the process of moving towards actualizing this type of mission.

Sensor network design is the first technology development area. Nearly every system requires sensor data to determine state, identify faults, or plan control actions. Since the addition of sensors is costly, it is important that sensor networks are optimally designed. There are many open problems to be researched, and while some of the research must be conducted in specific areas (i.e., how to make a sensor that measures X, or how to make this sensor more accurate/robust/reliable), there are many areas that are common to design of sensor networks of any type.

The next common technology is **System Health Management**. This includes fault detection and isolation, but also includes some understanding of the environment and the system model as a whole. This technology is already being developed for robotic spacecraft, but must be scaled to the level of complexity that the human spacecraft will have. **Planning** is another common technology, and will have a distributed yet hierarchical structure similar to System Health Management. Planning can take on many meanings, but the ability to order, distribute, and execute tasks is needed for any closed loop autonomous system.

The **verification and validation of autonomous systems** is an area where tremendous development is needed. The methods of testing these new types of systems will likely be completely new and different from current processes, and the development of these methods must happen concurrently with the development of the technology. This approach is recommended for integrating autonomous systems technology into the spacecraft for the same reasons as the integrated co-development of the autonomous control system with its V&V methods is recommended here. The focus on how the system will be tested will influence the design of the overall autonomous system, and it is likely that this integrated co-development will see reduced development time and cost overall than a series or waterfall development method.

Finally, the **situational awareness of ground controllers** is a technology development area that all subsystems will encounter. This is based on the need to collect, select, and shape data appropriately so that the ground controller quickly understands the situation on the spacecraft. This data will include sensor data (perhaps in a processed or analyzed form), plan information, and recommendations from the autonomous system. The driving constraints will be the latency and downlink bandwidth allowed; information will need to be carefully designed for maximum insight into the situation. All subsystems in the spacecraft will need to follow these principles and adhere to the design and interface of this data for the situational awareness of the ground controllers, and so it is an area of research broadly applicable to the system. A related topic to this is the transparency of autonomous systems, which is a broad interest in autonomous systems development. For example, there is an IEEE Standards Working Group exploring the creation of testing standards for autonomous systems on system transparency- this will allow for insight into the autonomous systems' execution to inform a variety of stakeholders.

4.2.2 Investment Priorities

The top investment priorities are conducting the recommended architectural trades, data management, sensor networks and technologies, robotics capabilities, and distributed autonomous commanding. The architectural trades recommended in this document are important investments because careful consideration of these problems will guide the design of the spacecraft and the technology investments that are needed up to a decade in advance. For example, self-healing materials versus robotic repair is part of a trade that needs to be considered. Both technologies are fairly low readiness level, but are established enough that a detailed trade is possible. Spacecraft architecture and funding priorities would change if one of these options (or a third option) are pursued.

Data management is central to any autonomous system. Investments in this area can be focused to NASA needs (due to low bandwidth and latent communications) and are essential to all types of autonomous systems. Without clear advances in the state of data collection, storage, processing, annotation, and analysis, the mission described in this report will not be possible. Sensors are the other side of this same coin. Data collection must start with some sensing mechanism connected to a network or data collection device. These sensors are costly (mass, power, processing, etc.), and are required for autonomy in the absence of the crew. The situational awareness afforded by having a crew on board will have to be replaced, and sensors will be the start of a chain of technologies needed to do this. Every subsystem described in this document requires sensing. Advances must be made to ensure that the data demands are met.

Robotics capabilities are repeatedly mentioned as a solution to autonomy gaps throughout this document. The state of robotics technology is not yet at the level that makes it easy to commit to robotic agents as critical systems in the spacecraft management. Investment is required to bring robotics technology to the necessary levels, as the benefits of robust mobile manipulators and inspectors will likely far outweigh the initial cost of technology investment. Like the two technologies in the previous paragraph, robotic agents are likely to bring game changing capabilities that will enable maintenance of an uncrewed human spacecraft.

Finally, distributed autonomous commanding will be important to allow for the various configurations and environments in which the spacecraft will be operated. The spacecraft will be close to the Earth and far away, and it will have humans on board and it will be uncrewed. The commanding will be shared between ground controllers, crew, and the spacecraft, likely in different proportions in different mission phases. Also, there is plenty of reasoning and control that can be conducted local to subsystems before a centralized spacecraft autonomous controller (VSM) would need to get involved. The many different entities that may be controlling the spacecraft at various times requires careful design of the autonomous control architecture, and this work must be invested regardless of the specific autonomous functions that will be executed as part of the framework.

Individual subsystems have prioritized the autonomous function investments that are important to them. These priorities can be found in Appendix D.

5.0 Conclusions

In conclusion, independence of the spacecraft from ground control will be essential for missions beyond low Earth orbit because of the communications constraints that are imposed by physics and the resources available. Independence of the spacecraft from crew will be essential for missions that include planned uncrewed times, such as the design reference mission described here. Further need for autonomy is driven by the criticality of the health of the spacecraft to the overall mission and crew- this spacecraft must be ready to bring the crew home, and extremely limited ability to resupply or repair the spacecraft due to unplanned events will exist in this scenario. So, in all, there is overwhelming data that the technology that provides this independence, described throughout as autonomous functions or "autonomy," is the only option for success. In particular, the primary driver for the need for autonomy is the time to criticality. Events that cannot be handled within the time frame dictated by command latency or by the time needed to downlink the necessary amount of data in the bandwidth available MUST be handled autonomously by the spacecraft, either in full or in part.

System architecture and design are critical to the autonomy of the spacecraft. Design and architecture decisions must be made at the initial stages in order to enable autonomous functionalities that are needed by the spacecraft. These design decisions could be to reduce the complexity of the system (i.e., remove all toxic gases from the system design, which removes entire subsystem functions like detection and scrubbing), to reduce the interaction between subsystems (i.e., design thermal properties such that no maneuvering is needed for the active TCS), or to enable situational awareness of the overall system (i.e., sensor arrays and associated processing). The focus of system design, coming back to the previous point, should be to do what is necessary to reduce the time to criticality for each subsystem and each function. The more system design can accomplish that, the less that the subsystem autonomy and VSM will have to do for the spacecraft. Reduction in complexity of the autonomous algorithms and functions that will need to be developed, tested, and flown will have direct benefits in the cost and complexity of building and operating the spacecraft.

Finally, many of the contingency and maintenance procedures used on the ISS today have a lot of reliance on the crew. While improvements in robustness and reliability of subsystems and components are expected, it is naïve to believe that no failures, emergencies, or unexpected events will happen to the spacecraft during its uncrewed phase. As such, these contingencies must be handled in one or more ways. First, robotic agents for sensing, manipulation, and mobility could be used as a direct replacement for crew functionalities, which allows for a common spare approach, fewer sensors to enable coverage, and a lower risk posture for the overall mission. For some types of contingencies (namely those that result in component failures), a "spares-in-place" strategy could also result in a lower risk posture for the mission, as long as there are enough in situ spares to accommodate for the number needed based on the criticality of the component. This appears in many places in this work in discussions of sizing tanks or considering the spare sensors needed in arrays to cover failed sensors. A third option is to increase the risk posture of the mission temporarily until the crew is able to fix the problem. For example, if two redundant strings of a subsystem are needed, but one fails, an increase to the risk posture would allow the crew to return to a degraded spacecraft and operate that spacecraft for some amount of time before repairs can be completed. In any case, mission and spacecraft designers must decide what the appropriate strategy is (possibly on a subsystem by subsystem basis) and account for it in the overall design.

Autonomy is essential to the success of future exploration missions, and autonomous systems come in many forms. Careful consideration of autonomous functions that need to be performed by a spacecraft must be completed early in the mission design in order to create a system that is capable of being operated in the conditions required. The operational mindset must be built into spacecraft that are going beyond low Earth orbit. This document attempts to give essential information to those who are tasked with these mission architecture and system design efforts.

Appendices

A. Autonomy and the Transition from Dormant to Crewed Operations

Stan Love and Jeremy Frank

This study has focused on making human-tended space systems that can operate and maintain themselves in a dormant state with minimal interaction with human flight controllers. By definition, the dormant state ends when a crew arrives. So does the applicability of this study. Nevertheless, to guide future thinking about transitions between dormant and active states, and how this transition changes the characteristics of autonomy-enabling technology, it makes sense to briefly discuss how the operation of a largely autonomous system might have to change when company arrives.

A.1 Fault Detection and Recovery

First the consequences of the arrival of the crew on fault detection and recovery is discussed. If a robot spacecraft encounters a problem it can't solve on its own, it can automatically enter a protective "safe mode" that terminates all but the most essential functions and places the system in a state where it is thermally stable, able to receive commands from Earth, and gaining (or at least not losing) electrical energy. Safe mode maximizes the length of time the system can survive while waiting for engineers on Earth to solve the problem and transmit a solution.

With a crew on board, automated failure responses must protect not just the spacecraft, but the crew as well. The definition of "essential function" expands when a crew is present. Even in safe mode, the life support system, interior lighting, and other equipment must remain operational in order to meet the crew's needs. The life support system is a major consumer of energy and producer of heat, so the spacecraft's power and thermal margins will shrink. The extra equipment operating when the crew is present means that the number of malfunctions that automation must contend with is larger. Furthermore, the number of critical malfunctions that could threaten the crew, the spacecraft, or both, is greatly increased. The nature of automated responses changes as well; the system might have to sacrifice other spacecraft functions in order to maintain a habitable environment for the crew for as long as possible.

A.2 Nominal Operations

Beyond fault detection and response, nominal operations also change when the crew is present. Repetitive nominal operations lend themselves well to ‘scripting,’ which autonomy technology can usually handle. But unexpected events such as micrometeoroid strikes and solar flares require special handling. The crew itself will cause unexpected events that influence how autonomous and automatic systems function. The astronauts' human metabolic cycles will vary on many timescales. Their use of logistics will be hard to predict. They are likely to change the order of planned system activities. All of these factors have consequences. The meaning of "nominal operations" for spacecraft systems also changes. More frequent communication, different power and thermal duty cycles, and more dynamic operations such as spacewalks and vehicle arrivals and departures all contribute to a new normal.

A.3 Robotic Operations

If robots operate during dormant time periods, whether autonomously or via teleoperation, they will face fewer operating constraints. Robots doing indoor work won't have to avoid humans in their work envelope. Robotic work outside the habitat can proceed around the clock without disturbing anyone's sleep. Those and other constraints to robotic operations return when the crew does.

A.4 System Benefits of an Onboard Crew

The presence of a crew complicates autonomous operations, as described above. But a local human operator can provide benefits as well. A computer may not know the date and time after a power cycle, but a human crew member will. Humans are sensitive to burning odors that might verify a short circuit, and rattling sounds that might presage the failure of a pump or fan. They can determine the right way to point the solar arrays by looking out the window.

Humans can augment automation in more ways than just using their senses. They can direct robots to perform tasks more effectively and more quickly than a distant control center can. "Flexible" human intelligence is sometimes able to solve complex problems that stymie "brittle" automated algorithms. Although astronaut crews are necessarily limited in numbers and expertise compared to flight control teams, they should have enough smarts (or enough onboard procedures) to recover the ship after an upset without having to contact Mission Control. This capability is especially valuable if a malfunction occurs in the communication system. On the other side, software can augment humans by performing high-speed calculations, reacting more quickly than humanly possible, and taking action while the crew is asleep or incapacitated.

A.5 Design Implications for Autonomy With and Without Crew

Making the best use of autonomy-enabling technology in concert with a crew demands attention from algorithm designers, developers, and user experience specialists. The required capability of the autonomy-enabling algorithms that operate the ship and respond to failures in the dormant state are likely to be similar to those needed when a crew is present. But the *output* of those algorithms will be used differently by the human crew than by the dormant spacecraft.

A well-designed user interface (including cockpit displays and controls) is a necessity. Volumes have been written on this topic, but the central ideas are to present information in a way that highlights the important, omits the trivial, minimizes the chance for misinterpretation, and accepts commands or inputs from the crew with minimal chance of error. The user interface should notify the crew when the spacecraft state changes because of an action by the software. Some critical actions will require the crew to grant permission for the system to perform.

In a crewed ship, automated reasoning algorithms (such as those that govern planning, system management, failure response, and robotic operations) should be able to offer alternative courses of action with a recommendation of which to pursue. They should be able to accept the crew's added information or direction ("Open the pod bay doors, Hal"), to run projections of future system behavior for evaluating "what-if" scenarios, and to provide explanations for actions taken or conclusions reached.

Implicit in this discussion is thoughtful allocation of tasks among human operators and autonomous systems. This is an open area of study, requiring both a concrete system and considerable analysis.

In closing, a full treatment of the interaction between astronauts and onboard automation is beyond the scope of the present study, which examines the dormant state. The preceding paragraphs outline a few of the major considerations that a follow-on investigation, focused on the active state, would have to explore.

B. Ground Operations for Dormant Spacecraft

“Planning for Failures”: Justification for Ground-based Data Analysis

When a spacecraft is enabled for autonomy, the goal should be full autonomy leveraging the knowledge of the engineering and operations teams. As knowledgeable as the teams are, “known-unknowns” and “unknown-unknowns” will continue to surface at all levels of the spacecraft and with the space environment. This being the case, operations and engineering will need to transition from issue monitoring and resolution to issue prediction and pre-emption in order to cover gaps in the team’s knowledge. In order to transition to this new approach, the following will need to be considered:

There is a need for operational systems and capabilities to identify faults pre-emptively and update the software accordingly. The heart of the problem is identification of issues that have not manifested themselves directly or indirectly. This requires the ability to imagine spacecraft or space environment issues and rapidly identify or rule out a root cause using the following technologies. Advanced high-fidelity simulation capabilities can provide readily accessible information and capabilities to engineering and operations teams. These simulation capabilities can be a mix of hardware and software, however the hallmark is quick, ready access to high fidelity simulation without having to compete for time. Rapid turnaround on the simulator is needed to “fly” off-nominal mission scenarios. This does not require a high-fidelity simulation, but does require the inclusion of all subsystems in the simulation.

Analysis to identify what could go wrong next is a concept that uses data acquired from the dormant spacecraft and “plays it forward” in an attempt to predict what assets, components, subsystems, or systems are in line to fault or fail. If the analysis results in a positive identification, the scenario could be played out to determine if the flight software and data is capable of resolving the issue or even preventing the issue from occurring on the spacecraft.

Analysis of “unknown-unknowns” tests or challenges the veracity of the operations and engineering team knowledge, exploring failure options for simulation and analysis. This may be the most difficult concept to implement, requiring a significant amount of creativity and imagination. As candidate ideas are developed, simulation is used to test those ideas and determine if there are gaps in the spacecraft procedures or data.

Spacecraft level analysis based on a holistic approach ensures effects on all spacecraft subsystems are evaluated. Simulation capability must provide operations and engineering with the ability to assess fault, failure, and issue impacts across the spacecraft, identifying those areas requiring more resiliency.

C. Top 10 Reasons Why Dormant Spacecraft for Crewed Operations Are Different Than Robotic Planetary Exploration Spacecraft

Reason #10

Active Fire Suppression Systems. Fire suppression systems are a must since there is an oxygen-based atmosphere in the spacecraft. In dormant flight, the tendency is to assume the spacecraft can maintain an atmosphere that is not conducive to fire. However, there are dormant phases where increased levels of oxygen are required when a) the crew is returning from the surface or a fresh crew is arriving, b) the crew on surface relies on the spacecraft as a contingency habitat in case something goes wrong on the surface, and c) the spacecraft transitions into the hibernation phase after the crew departs.

Reason #9

Autonomous Spacecraft Operations. Significantly greater reliance on software to control and operate the spacecraft with little to no intervention from the ground or crew is required. The spacecraft is expected not only to identify faults and failures, but to recover flight assets as a result, all without engaging the crew or ground controllers. Although there may be a tendency to want to fault down into levels of "safe modes," the spacecraft's intelligent systems need to safely recover as many lost assets and restore its subsystems as it can before engaging the ground.

Reason #8

Shift in Operations Mindset. With time delays, ground operations and engineering must shift from "reactive" to "proactive" operational support. This requires the operations and engineering teams to think in terms of failure prevention and pre-emptive actions rather than "hands on" operation. Autonomous spacecraft operations will handle the day-to-day spacecraft operations through the software system.

Reason #7

Contingency Resources for Human Survival. This recognizes the potential need of the spacecraft to extend its mission timeline and support additional crew or extend crew stays. Human consumables are to be maintained and replenished by regular or timed logistics missions over the spacecraft's lifetime with ample supplies for contingencies. Unlike robotic spacecraft, crewed spacecraft must factor in the human element.

Reason #6

Serves as a "Life Boat" for the Crew. With humans in the equation, the time needed to rescue a crew will increase significantly. It will be up to the spacecraft to maintain a crew for extended periods of time until they can be relieved. With the crew involved, the basic equation is changed from accomplishing the mission on a robotic spacecraft to ensuring crew survival on a crewed spacecraft.

Reason #5

Automated Spacecraft Mission Reconfiguration. Dormant spacecraft may be reconfigured to accomplish many missions over their service life. Assets in orbit around a planet are designed for human habitation and are only bound by human imagination.

Reason #4

Automated Integration with Other Spacecraft. The spacecraft must be able to integrate with logistics flights and crewed transport flights over its service life without intervention or help from the ground. At times the spacecraft may double in size to support the mission as defined or integrate other permanent elements to support a greater human presence.

Reason #3

Autonomous Re-fueling. If the service life of the outpost spacecraft is to be extended, there has to be a capability to automatically re-fuel, replenish consumables, and maintain a state of stability over long periods of time without a crew.

Reason #2

Robotic Servicing and Maintenance. Without the crew on board, maintenance and servicing tasks will require robotics. Robots will also be used to perform repair and replacement work previously assigned to humans. Robotic capability will fill the gap to ensure the spacecraft is a viable concern for crews in transit to the spacecraft and for future missions.

…. And the #1 Reason

Maintenance of Multiple Ecosystems. Although this seems like a bit of a stretch, it is a fact. The subsystems on a dormant spacecraft must ensure the environment is conducive to human habitation, minus as much of those nasty bacteria as possible. Plus, there may even the levels of habitation depending on mission phase. For example, a dormant spacecraft may maintain the environment with minimal partial pressures if no crew is expected for several months. This environment differs from the environment that must be maintained if a crew is on the surface and contingency scenarios require expeditious return.

D. Subsystem Investment Priorities

D.1 Avionics

1. C&DH System Architecture Designs for Future Spacecraft
2. Deterministic Networking and Fault Tolerance
3. Radiation hardened or tolerant components
    a. Memory Storage
    b. High Speed Interconnects for Onboard Spacecraft Data Networks
    c. Single Board Computers

D.2 Communications

1. Networked, standards-based communications for EVAs, free-flyers, and space vehicles.
2. Validation of space-to-space communications with radiometric tracking.
3. Autonomous re-configuration of onboard communications and ground stations.

D.3 Environmental Control and Life Support System

1) Manual processes and activities (sampling, analysis, recovery and maintenance activities)
    a) Do these need to be performed when crew is not present?
    b) How can these processes and activities be automated?
        i) Robotic performance of crewed activities
        ii) In-line sensing, redundancy, etc.
2) Maintaining autonomous capability throughout the uncrewed period
    a) Sensor operational life and reliability
    b) Reliability of mechanical components that degrade with non-use, such as electronics, seals
3) Automation associated with on-orbit implementation of ground processes (may be desired, not required)
    a) Sensor calibration
    b) Chemicals mixing (e.g. urine pretreat)
    c) Silver biocide dosing of potable water

D.4 Software

1. Intelligent Processing Optimization. Optimal distribution of intelligent agents or intelligent processing amongst the spacecraft subsystems to enable autonomy within each subsystem, i.e. aspects of “distributed VSM” to work in conjunction with the VSM.
2. Data Design Methodologies. Rapid development of data interrelationships and interdependencies for all data at the spacecraft level, the subsystem level, and in between.
3. Data Analysis/Deduction. Techniques and methods to identify alternate data values to mitigate incomplete or corrupt data in or between subsystems and to identify faulty subsystem components (e.g. sensors).
4. Variable Content Downlist. Capabilities to target specific downlist data depending on the spacecraft situation in order to handle spacecraft states ranging from nominal operations to in-depth fault/failure management.
5. Flexible Avionics Hardware Architectures. Avionics hardware topologies that allow for seamless fault-down and recovery without loss of data or time critical processes, all the while minimizing weight and power.

E. Acronyms

ACS	Attitude Control System
AR&D	Autonomous Rendezvous and Docking
ATCS	Active Thermal Control System
BIT	Built-In Test
C&DH	Command and Data Handling
C&W	Caution and Warning
CIS	Cognitive Information Synthesis
CMG	Control Moment Gyro
DSG	Deep Space Gateway
ECC	Error Correction Code
ECLSS	Environmental Control and Life Support System
EDL	Entry, Descent, and Landing
EVA	Extravehicular Activity
FDIR	Fault Detection, Isolation, and Recovery
FMEA	Failure Modes Effects Analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IMV	Inter-module Ventilation
ISS	International Space Station
IVA	Intravehicular Activity
LOC	Loss of Communications
MTBC	Mean Time Between Commands
MTBI	Mean Time Between Interference
MTBT	Mean Time Between Telemetry
NASA	National Aeronautics and Space Administration
NPR	NASA Procedural Requirement
PCS	Pressure Control System
PMAD	Power Management and Distribution
PPB	Propulsion and Power Bus
PVT	Pressure-Volume-Temperature
RCS	Reaction Control System
RF	Radio Frequency
RPOD	Rendezvous, Proximity Operations, and Docking
R&R	Remove and Replace
SEU	Single Event Upset
SHM	Structural Health Management
SOA	State of the Art

SSRMS	Space Station Remote Manipulator System
TCS	Thermal Control System
TRL	Technology Readiness Level
TTC	Time to Criticality
UHF	Ultra High Frequency
VHDL	VHSIC Hardware Description Language
VHF	Very High Frequency
VHSIC	Very High-Speed Integrated Circuit
VMC	Vehicle Management Computer
VSM	Vehicle Systems Manager
V&V	Verification and Validation