Hazard Analysis of Complex Spacecraft using Systems-Theoretic Process Analysis [*]

Takuto Ishimatsu[†], Nancy G. Leveson[‡], John P. Thomas[§], and Cody H. Fleming[¶]
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Masafumi Katahira[#], Yuko Miyamoto[**], and Ryo Ujiie[††]
Japan Aerospace Exploration Agency, Tsukuba, Ibaraki 305-8505, Japan

Haruka Nakao[‡‡] and Nobuyuki Hoshino[§§]
Japan Manned Space Systems Corporation, Tsuchiura, Ibaraki 300-0033, Japan

Abstract

A new hazard analysis technique, called System-Theoretic Process Analysis, is capable of identifying potential hazardous design flaws, including software and system design errors and unsafe interactions among multiple system components. Detailed procedures for performing the hazard analysis were developed and the feasibility and utility of using it on complex systems was demonstrated by applying it to the Japanese Aerospace Exploration Agency H-II Transfer Vehicle. In a comparison of the results of this new hazard analysis technique to those of the standard fault tree analysis used in the design and certification of the H-II Transfer Vehicle, System-Theoretic Hazard Analysis found all the hazardous scenarios identified in the fault tree analysis as well as additional causal factors that had not been) identified by fault tree analysis.

I. Introduction

Spacecraft losses are increasing stemming from subtle and complex interactions among system components. The loss of the Mars Polar Lander is an example [1]. The problems arise primarily because the growing use of software allows engineers to build systems with a level of complexity that precludes exhaustive testing and thus assurance of the removal of all design errors prior to operational use [2,3] Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA) were created long ago to analyze primarily electro-mechanical systems and identify potential losses due to component failure. They are based on reliability theory. Our new complex, software-intensive designs, however, need more powerful analysis approaches that go beyond component failure to identify additional causes of accidents. While engineers have tried to extend these traditional techniques in ad hoc ways, the identification of causal scenarios involving subtle and unidentified interactions among components and system design errors (including requirements errors) are not easily handled by these techniques, if at all. The ad hoc nature of the extensions means that there is no way to systematically ensure that cases have not been omitted. The problem is that the underlying accident causality model of these techniques, which assumes that accidents are caused by chains of directly related component failure events, does not include all the types of accident that we are now having in our software-intensive systems and the types of flawed human decision making involved.

This paper describes a new hazard analysis technique, called Systems-Theoretic Process Analysis (STPA), which is based on concepts in systems theory and control theory. STPA identifies the traditional causes of losses identified

[*] Preliminary results of the analysis in this paper were presented at the IAASS Conference in 2010 and 2011. Additional information has been included in this paper that has not previously been published.

[†] Graduate Research Assistant, Department of Aeronautics and Astronautics, 33-409, AIAA Student Member

[‡] Professor, Department of Aeronautics and Astronautics, Engineering Systems Division, 33-334

[§] Graduate Research Assistant, Engineering Systems Division, 33-407, AIAA Student Member

[¶] Graduate Research Assistant, Department of Aeronautics and Astronautics, 33-407, AIAA Student Member

[#] Senior Engineer, JAXA's Engineering Digital Innovation Center, 2-1-1 Sengen, Tsukuba, Ibaraki 305-8505, Japan

[**] Associate Senior Engineer, JAXA's Engineering Digital Innovation Center, 2-1-1 Sengen, Tsukuba, Ibaraki 305-8505, Japan

[††] Engineer, JAXA's Engineering Digital Innovation Center, 2-1-1 Sengen, Tsukuba, Ibaraki 305-8505, Japan

[‡‡] Associate Senior Engineer, Safety and Product Assurance Department, 1-1-26 Kawaguchi, Tsuchiura, Ibaraki 300-0033, Japan

[§§] Senior Engineer, Safety and Product Assurance Department, 1-1-26 Kawaguchi, Tsuchiura, Ibaraki 300-0033, Japan by FTA and FMEA, but it also identifies additional causes. The technique works on an engineering model of the system and has well-defined steps, which are potentially at least partially automatable. The next two sections describe STPA and the underlying extended causality model, called STAMP (Systems-Theoretic Accident Model and Processes), on which it is based. The new analysis technique is demonstrated by its application to the JAXA (Japanese Aerospace Exploration Agency) H-II Transfer Vehicle (HTV). The results of the FTA used to certify the HTV are compared with the results of STPA on the same design.

II. STAMP: An Extended Accident Causation Model

Accidents, which in STAMP are defined very generally as unacceptable losses, have traditionally been conceived as occurring from a sequence of directly related failure events, each of which leads to the next event in the chain of events. With increased system complexity and the introduction of software, which does not "fail" in the sense that hardware does, new accident processes are arising. STPA (System-Theoretic Process Analysis) is built on the foundation of an extended accident model that includes the traditional chain-of-failure-events model but extends it to include losses arising from system design errors, software requirements errors, system engineering flaws, and organizational and managerial deficiencies.

In STAMP [4], accidents are complex processes that include a chain of failure events as a subset. Systems are conceived as being a collection of interacting control loops. In this causality model, losses result from a lack of adequate control over the behavior of the system components rather than simply component failures. There is a set of constraints related to the behavior of the system components (physical, human, or social) that enforces the desired system property. Losses occur when the behavior of the components or the interactions among the components violate these constraints. The constraints are enforced by controls and controllers.

Thus system safety in STAMP is treated as a dynamic control problem rather than a component failure problem. For example, the O-ring did not control the propellant gas release by sealing the gap in the field joint of the Space Shuttle Challenger. The software did not adequately control the descent speed of the Mars Polar Lander [1]. The development process did not adequately control the generation of the load tape in the 1999 Titan IV/Centaur/Milstar loss [5].

The concept of *control* is used very generally in STAMP. Component failures and unsafe component interactions or behavior may be controlled through design (e.g., redundancy, interlocks, fail-safe design) or through process, in which the controls may include the development processes, manufacturing processes and procedures, maintenance processes, and operations. Control may also be implemented by managerial, organizational, or social controls.

In STAMP, emphasis is changed from "prevent failures" to "enforce safety constraints on system design". Losses are not simply the result of an event or a chain of events but involve a complex, dynamic process. The events themselves result from a lack of enforcement of safety constraints in system design and operations. If the larger, dynamic process is not considered, potential causes of losses may be missed in the design and analysis process and be uncontrolled in design and operations.

In general, losses in STAMP occur in three ways. The first way is that the control structure or control actions do not enforce the safety constraints, resulting in unhandled or uncontrolled component failures, unhandled environmental disturbances or conditions, or unsafe interactions among system components. The second occurs when control actions are inadequately coordinated among multiple controllers. The third is that the control structure degrades over time and becomes inadequate.

There is one other important concept in STAMP, which is the role of process/mental models in losses. Fig. 1 illustrates the basic systems theory concept that every controller must contain a model of the system it is controlling. In humans, the process model is usually called a mental model. The process model is used by the controller to determine what control actions to implement. Accidents, particularly those involving software and human errors, often occur because the process model becomes inconsistent with the actual state of the controlled process causing incorrect control actions to be issued. For example, in the Mars Polar Lander (MPL), the software thought that the spacecraft was on the surface of the planet and shut off the descent engines. Process models are kept consistent with the process state either through prediction (feedforward control) or through feedback (feedback control). The concept of process model inconsistency with the actual controlled system provides a much better explanation of software or human control errors than conceiving of them as random component failure.

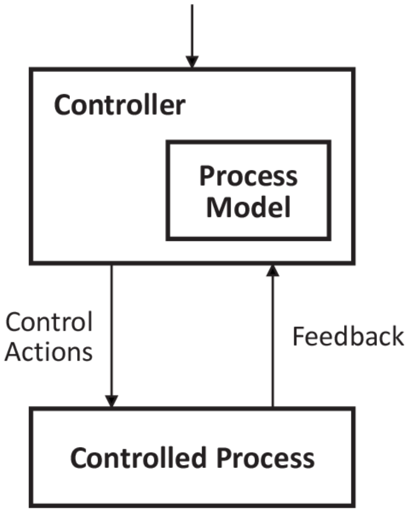


Fig. 1 The role of the process model in a control loop.

How do the process models become inconsistent with the state of the controlled systems? They may be wrong from the beginning, there may be missing or incorrect feedback, the algorithm being implemented by the controller may update the models incorrectly, or time lags may not be properly accounted for. The result may be uncontrolled process states, uncontrolled disturbances, or inadvertently commanding the system into a hazardous state.

The causes of an accident in STAMP listed earlier can then be augmented with four types of unsafe control actions, i.e., accidents occur when the controllers' process models do not match the actual state of the process and 1) a control action required to avoid a loss is not provided or is not followed, 2) an unsafe control action is provided, 3) a potentially safe control action is given at the wrong time (too early, too late, or in the wrong sequence), or 4) a control action required for safety is stopped too soon or is applied too long (for nondiscrete commands).

STPA is a hazard analysis method based on the STAMP accident causality model. It starts from fundamental system engineering activities, including the identification of losses or accidents to be avoided, the hazardous behavior that could lead to these losses, safety requirements and constraints, and the basic system control structure used to avoid these losses. The primary goal of STPA is to generate detailed safety requirements and constraints that must be implemented in the design in order to prevent the identified unacceptable losses. It achieves this goal by identifying unsafe control behavior and the scenarios that can lead to this behavior, including component failure scenarios.

The goal of STPA is the same as traditional FTA, but STPA includes a broader set of potential scenarios including those in which no failures occur but the problems arise due to unsafe and unintended interactions among the system components. STPA also provides more guidance to the analysts than FTA. Functional control diagrams and a set of generic causal factors are used to guide the analysis. Fig. 2 shows the basic causal factors used in an STPA analysis of a functional control diagram. The next section describes the STPA process in more detail using a case study of its application to the JAXA HTV unmanned cargo spacecraft.

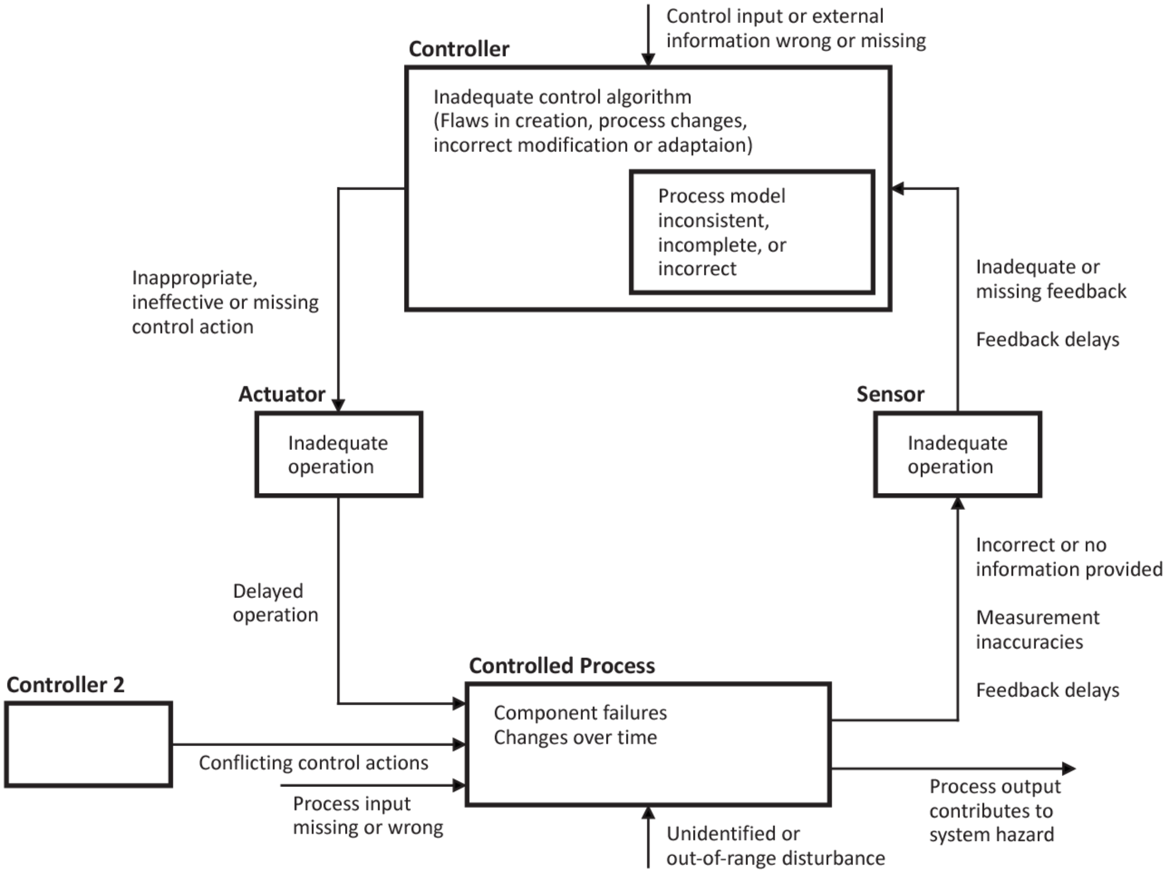

Control input or external information wrong or missing
Controller
Inadequate control algorithm (Flaws in creation, process changes, incorrect modification or adaptaion)
Process model inconsistent, incomplete, or incorrect
Inappropriate, ineffective or missing control action
Inadequate or missing feedback
Feedback delays
Actuator
Inadequate operation
Sensor
Inadequate operation
Incorrect or no information provided
Measurement inaccuracies
Feedback delays
Delayed operation
Controller 2
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Unidentified or out-of-range disturbance
Process output contributes to system hazard

Fig. 2 Basic causes of unsafe control.

III. Case Study: Hazard Analysis of JAXA HTV Using STPA

This section illustrates the use of STPA on a real spacecraft design. The relevant aspects of the spacecraft are first described and then the STPA analysis is presented.

The HTV is an unmanned cargo transfer spacecraft that is launched from the Tanegashima Space Center aboard the H-IIB rocket and delivers supplies to the ISS. In the development of the HTV, NASA safety requirements were applied, and potential HTV hazards were analyzed using FTA, with the emphasis being on the hazards of the integrated operation phase (i.e., approaching, berthing with, and departure from the ISS). The results of the FTA-based hazard analysis were documented in the hazard report. “Collision with the ISS” is the highest severity hazard and is the focus of the analysis shown in this paper.

The validity of all the contents of the hazard report was reviewed by the NASA Safety Review Board, such as the identified hazard causes, the hazard controls used for the causes, the design of the control, and the verification method. NASA and JAXA also analyzed the hazards identified for the ISS-HTV integrated operation and documented the results as the Integrated Hazard Analysis (IHA). In addition to redundant design for identified safety-critical components, a collision avoidance maneuver (CAM) was implemented to abort from the ISS collision trajectory if redundant components fail. In accordance with the results of the IHA, NASA and JAXA defined the flight rules for the integrated operation.

HTV operations can be divided into the following phases: (1) launch, (2) rendezvous with the ISS, (3) berthing with the ISS, (4) operations while berthed with the ISS, (5) undock/departure from the ISS, and (6) reentry. This paper focuses on the berthing phase, as shown in Fig. 3, because the ISS is involved in this phase and could be damaged by inadequate controls on the HTV final approach and capture operation and astronauts could be potentially injured or killed [6,7].

In Sec. II, three general causes of accidents in STAMP were identified. The first occurs when the control structure or control actions do not enforce the safety constraints, leading to unhandled or uncontrolled component failures, unhandled environmental disturbances or conditions, or unsafe interactions among system components. The STPA analysis for this type of loss is illustrated in Sec. II.B using the HTV capture operation (described in Sec. II.A). The second type of loss, control actions that are inadequately coordinated among multiple controllers, is illustrated by the HTV final approach phase. In this phase of operation, the HTV can be controlled by its own software, by the astronauts, by NASA mission controllers in Houston, and by JAXA mission controllers in Tsukuba. The potential exists for several of these controllers to inadvertently provide an unsafe combination of commands although each command may be "safe" by itself. Analysis of multiple controllers is described and illustrated in Sec. II.C. The third cause of accidents in STAMP, degradation of the control structure over time so that the safety constraints are no longer enforced, is primarily handled through operations and standard processes and is not considered further in this paper.

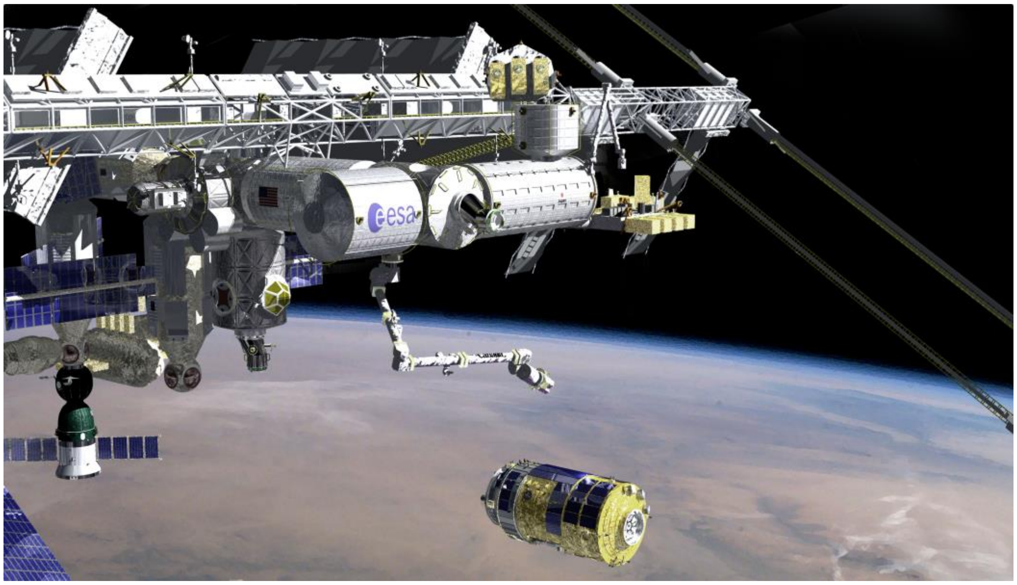


Fig. 3 HTV berthing with the ISS (image courtesy of JAXA) [6].

A. Overview of the HTV Capture Operation

After launch, the HTV performs an automated rendezvous flight to carry cargo to the ISS. As shown in Fig. 4, the HTV approaches the ISS from the nadir side of the ISS (R-bar approach). The HTV is then grappled by the station's robotic arm, called the SSRMS (Space Station Remote Manipulator System), and berthed to the ISS. The HTV approach sequence (called proximity or PROX operations) proceeds in four stages.

In the first stage, after final approach is given by the Mission Control Center at NASA's Johnson Space Center in Houston (MCC-H), the HTV Mission Control Room (MCR) at Tsukuba Space Center (TKSC) commands the HTV to begin the Approach Initiation (AI) Maneuver. The HTV moves from the AI point to the final approach point 500 m below the ISS guided by Relative Global Positioning System (RGPS) navigation.

In the second stage, while keeping its attitude relative to the ISS by using its attitude control system, the HTV begins its final approach using a laser sensor called the Rendezvous Sensor (RVS), beaming the laser to the reflector located on the nadir side of the Kibo module (RVS navigation).

The HTV holds its approach twice: at 250 m below the ISS (hold point) and at 30 m below the ISS (parking point). At the hold point, the HTV performs a 180-degree turn (yaw-around) to prepare for a Collision Avoidance Maneuver (CAM) in case of an emergency (for example, the HTV's relative position is too close or relative approach rate is faster than the predefined threshold).

Finally, once the HTV reaches 10 m below the ISS, called the Capture Box, the ISS crew disables the HTV thrusters by commanding the deactivation (*Free Drift*) and then manipulates the SSRMS to grapple the Flight Releasable Grapple Fixture (FRGF) of the HTV.

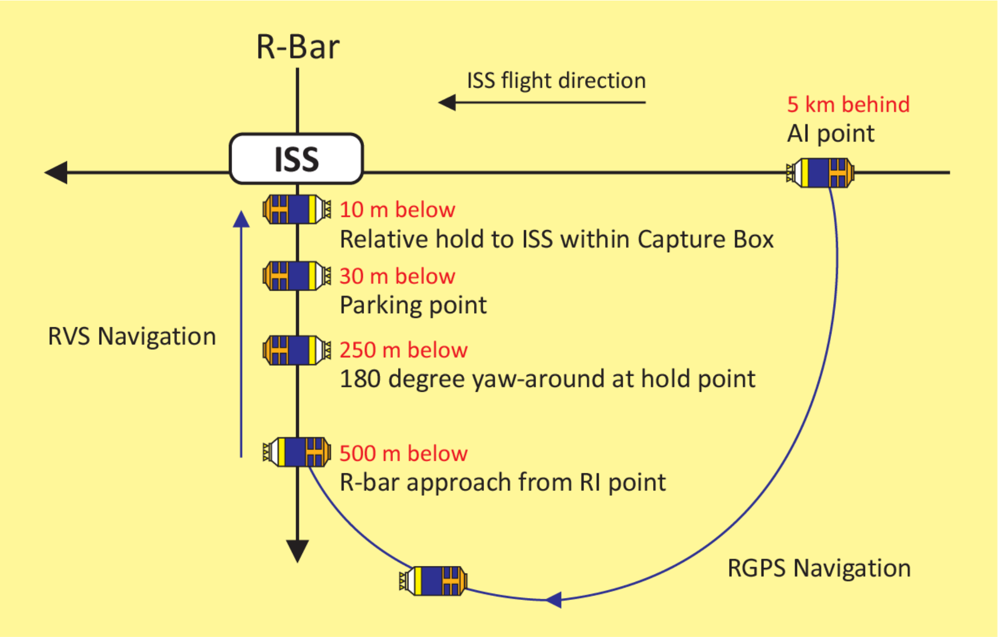


Fig. 4 HTV proximity operations [6].

During its final approach, the ISS crew can send commands to the HTV for immediate critical operations using the Hardware Command Panel (HCP) shown in Fig. 5. The HCP is deployed on the Robotics Work Station in the Cupola before the PROX Operations begin. The *Abort* function on the HCP moves the HTV away from the ISS. When *FRGF Set* is commanded, the FRGF is separated from the HTV (for example, to detach the HTV from the SSRMS in case the SSRMS cannot ungrapple the FRGF). *Retreat* causes the HTV to retreat to 30 m or 100 m below the ISS. *Hold* commands the HTV to hold its approach. Finally, in *Free Drift*, the HTV thrusters are disabled for the capture operation

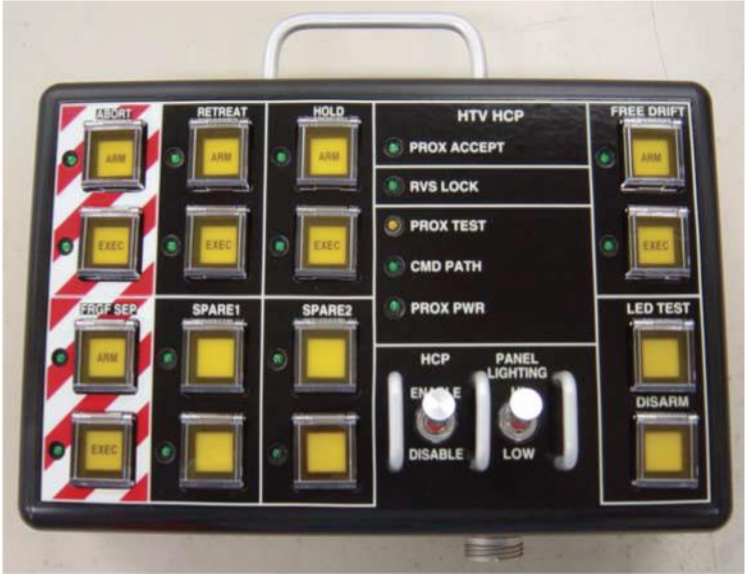


Fig. 5 Hardware Command Panel (HCP) (image courtesy of JAXA) [6].

STPA is illustrated on this procedure.

B. STPA Analysis

STPA can be divided into three steps: 1) identifying potentially hazardous control actions, 2) identifying scenarios that could lead to these control actions, and 3) identifying potentially unsafe interactions among multiple controllers.

1. *Identifying Potentially Unsafe Control Actions (STPA Step 1)*

The first goal of STPA is to identify potentially unsafe control actions and thus the safety constraints that must be enforced in spacecraft design and operations. STPA views hazardous states as a result of ineffective control. Therefore, the assessment proceeds by identifying the potential for inadequate control of the system that could lead to a hazardous state.

During the PROX Operations, the most serious accident is obviously a HTV collision with the ISS. It might not only result in loss of mission, but it could also lead to damage to the ISS modules or the SSRMS and potentially to ISS crew death or injury.

The functional control structure diagram is divided into two levels of abstraction in order to limit its complexity. Fig. 6 shows the Level-0 control structure diagram for the HTV capture operation. This control structure has five major components: ISS, HTV, NASA ground station (GS), JAXA GS, and Tracking and Data Relay Satellite (TDRS) as a backup communication system. Note that this diagram is a control model, not an architectural model. A functional control diagram is required because the analysis method is based on control theory and analyzes the functional behavior of the system and not just the physical structure.

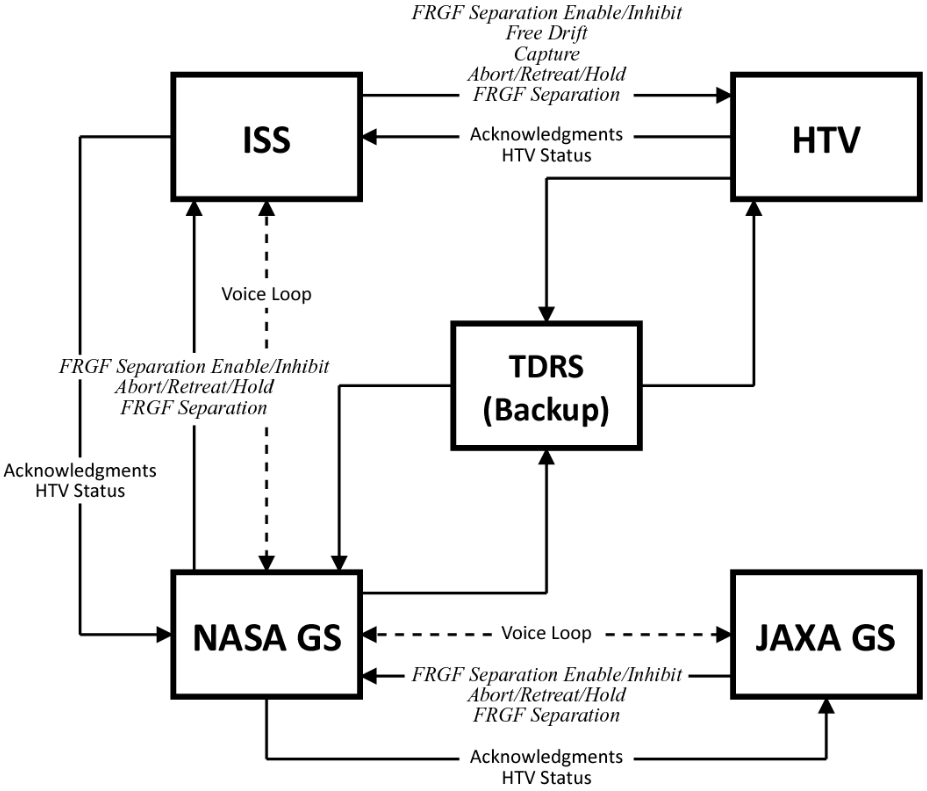


Fig. 6 Control structure for HTV capture operation – Level 0.

Fig. 7 shows the Level-1 ISS control structure (the box in the upper left hand side of Fig. 6). Major components inside the ISS include the Proximity Communication Command and Data Handling (PROX C&DH) system, the Hardware Command Panel (HCP), visual monitors/Portable Computer System (PCS), and the ISS crew. Connecting lines between those components show control actions, information, acknowledgments and feedback. There is also a voice loop connection between the ISSS crew, NASA mission control, and JAXA mission control.

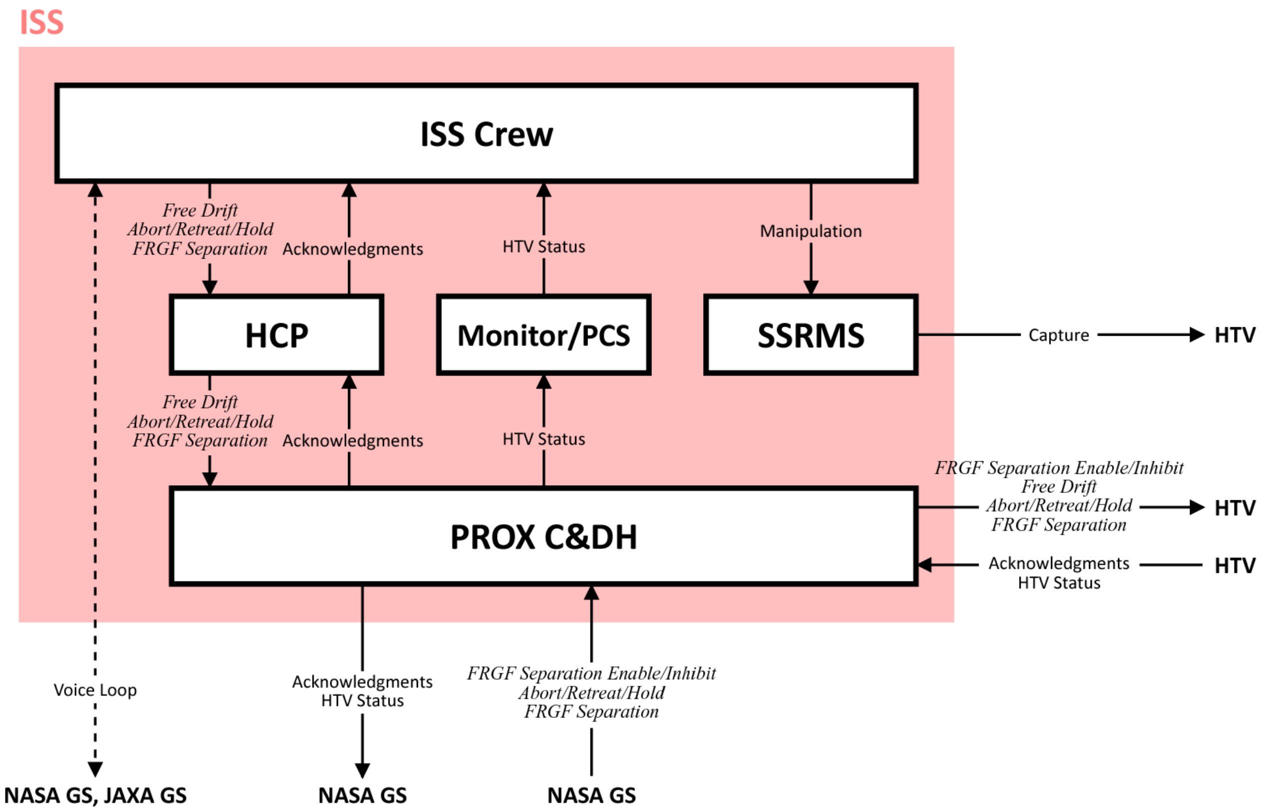


Fig. 7 Control structure for HTV capture operation – ISS Level 1.

Potential control actions are shown on the lines in the control structure diagram. For example, the ISS crew (Fig. 7) can issue *Free Drift*, *Abort/Retreat/Hold*, and *FRGF Separation* commands. Fig. 8 lists important control actions around the time of the capture. After the HTV has reached the Capture Box, the JAXA GS sends an *FRGF Separation ENABLE* command, which enables FRGF separation in preparation for an emergency. The ISS crew then sends a *Free Drift* command using the HCP to disable the HTV guidance and control functions. If the capture is started without this deactivation, the contact with the robotic arm could be interpreted as a disturbance by an external force, which would trigger an automatic attitude control action. Once the HTV has been deactivated, the ISS crew manipulates the SSRMS to grapple the HTV as promptly as possible. After the successful capture, the JAXA GS issues an *FRGF Separation INHIBIT* command to the HTV to prevent an unintended separation. These four events are the critical proximate events of the capture operation.

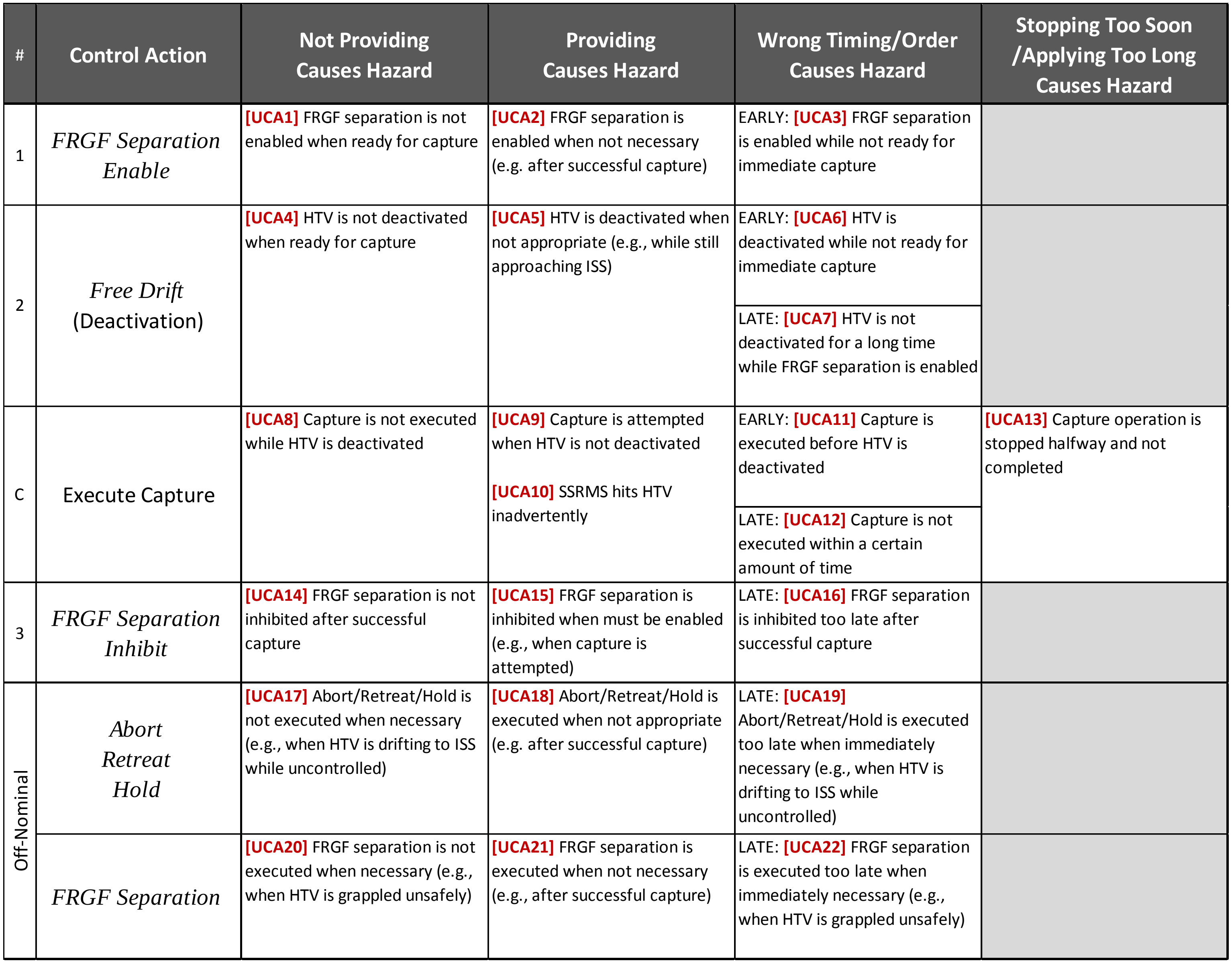
#	Control Action	from	to	Description
1	*FRGF Separation Enable*	JAXA GS	HTV	Enable FRGF separation in preparation for an emergency
2	*Free Drift* (Deactivation)	ISS (Crew)	HTV	Disable the HTV guidance and control functions
C	Execute Capture	ISS (Crew)	HTV	Manipulate the SSRMS to grapple FRGF of the HTV
3	*FRGF Separation Inhibit*	JAXA GS	HTV	Inhibit FRGF separation to prevent an unintended separation after the capture

Fig. 8 Control action sequence during capture operation.

For each control action, the conditions under which it could lead to a system hazard were identified using the four general categories of unsafe control actions (UCAs) as described in Sec. II: “not providing causes hazard,” “providing causes hazard,” “wrong timing/order causes hazard,” and “stopping too soon/applying too long causes hazard.” Although variants of this type of classification have been used in other hazard and reliability analyses, they usually stop with this information. The identification of the unsafe control actions in STPA is only the beginning of the analysis. The ultimate goal is to identify all the scenarios that can lead to these unsafe control actions. Fig. 9 shows the hazardous control actions identified. Because the ISS crew can issue *Abort*, *Retreat*, *Hold*, and *FRGF Separation* using the HCP, Fig. 9 includes these off-nominal commands in the analysis as well as the nominal control actions shown in Fig. 8. Each cell in the table shown in Fig. 9 describes the unsafe control actions, numbered from UCA1 through UCA22. The hazards to which each of these unsafe control actions could lead are summarized in Fig. 10.

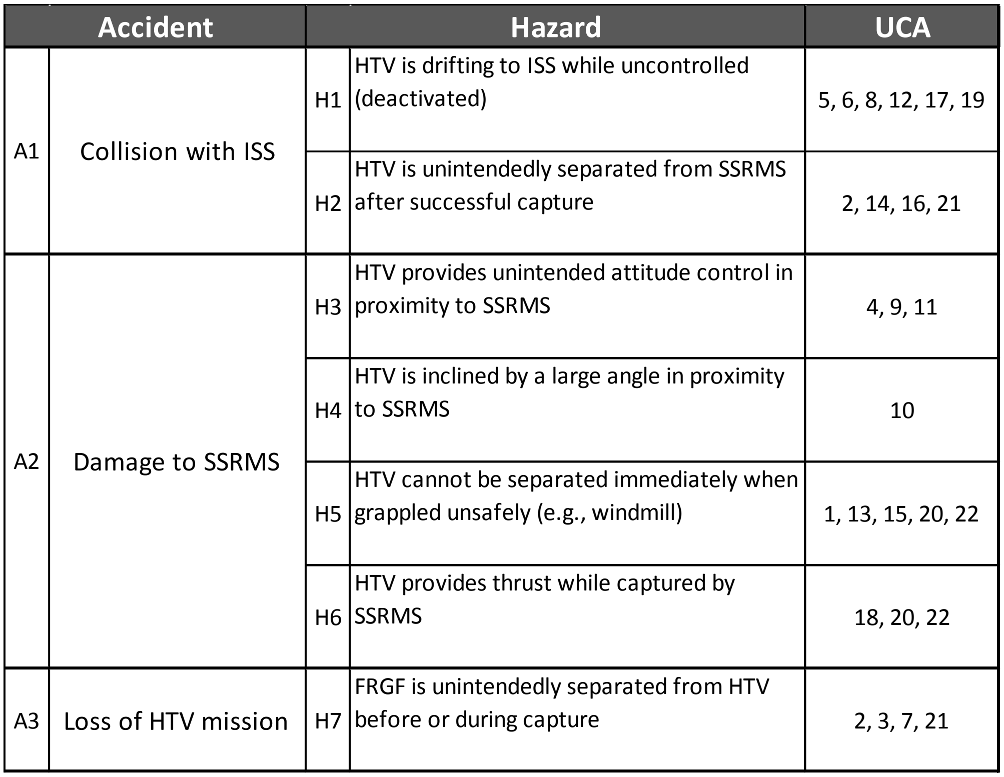
Once the unsafe control actions and related hazards have been identified, they can be translated into constraints (requirements) that must be enforced by the system design and operations. For example, if the capture is not executed early enough [UCA8, UCA12], the HTV will drift out of the capture box. In combination with no activation command or a late one [UCA17, UCA19], hazard H1 (HTV is drifting toward ISS while uncontrolled/deactivated) could occur. One safety constraint for this hazardous behavior is:

SC1.1: The ISS crew must activate the HTV appropriately within T seconds after drift out.



#	Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Wrong Timing/Order Causes Hazard	Stopping Too Soon /Applying Too Long Causes Hazard
1	*FRGF Separation Enable*	[UCA1] FRGF separation is not enabled when ready for capture	[UCA2] FRGF separation is enabled when not necessary (e.g. after successful capture)	EARLY: [UCA3] FRGF separation is enabled while not ready for immediate capture	
2	*Free Drift* (Deactivation)	[UCA4] HTV is not deactivated when ready for capture	[UCA5] HTV is deactivated when not appropriate (e.g., while still approaching ISS)	EARLY: [UCA6] HTV is deactivated while not ready for immediate capture LATE: [UCA7] HTV is not deactivated for a long time while FRGF separation is enabled	
C	Execute Capture	[UCA8] Capture is not executed while HTV is deactivated	[UCA9] Capture is attempted when HTV is not deactivated [UCA10] SSRMS hits HTV inadvertently	EARLY: [UCA11] Capture is executed before HTV is deactivated LATE: [UCA12] Capture is not executed within a certain amount of time	[UCA13] Capture operation is stopped halfway and not completed
3	*FRGF Separation Inhibit*	[UCA14] FRGF separation is not inhibited after successful capture	[UCA15] FRGF separation is inhibited when must be enabled (e.g., when capture is attempted)	LATE: [UCA16] FRGF separation is inhibited too late after successful capture	
Off-Nominal	*Abort Retreat Hold*	[UCA17] Abort/Retreat/Hold is not executed when necessary (e.g., when HTV is drifting to ISS while uncontrolled)	[UCA18] Abort/Retreat/Hold is executed when not appropriate (e.g. after successful capture)	LATE: [UCA19] Abort/Retreat/Hold is executed too late when immediately necessary (e.g., when HTV is drifting to ISS while uncontrolled)	
	FRGF Separation	[UCA20] FRGF separation is not executed when necessary (e.g., when HTV is grappled unsafely)	[UCA21] FRGF separation is executed when not necessary (e.g., after successful capture)	LATE: [UCA22] FRGF separation is executed too late when immediately necessary (e.g., when HTV is grappled unsafely)	

Fig. 9 Potentially hazardous control actions during capture operation.



Accident		Hazard		UCA
A1	Collision with ISS	H1	HTV is drifting to ISS while uncontrolled (deactivated)	5, 6, 8, 12, 17, 19
		H2	HTV is unintendedly separated from SSRMS after successful capture	2, 14, 16, 21
A2	Damage to SSRMS	H3	HTV provides unintended attitude control in proximity to SSRMS	4, 9, 11
		H4	HTV is inclined by a large angle in proximity to SSRMS	10
		H5	HTV cannot be separated immediately when grappled unsafely (e.g., windmill)	1, 13, 15, 20, 22
		H6	HTV provides thrust while captured by SSRMS	18, 20, 22
A3	Loss of HTV mission	H7	FRGF is unintendedly separated from HTV before or during capture	2, 3, 7, 21

Fig. 10 Accidents, hazards, and unsafe control actions.

2. Identifying Causal Scenarios for Unsafe Control Actions (STPA Step 2)

While some potentially unsafe behaviors can be designed out of the system without knowing all of the potential scenarios that can lead to those behaviors, more information about causality is usually needed. After the hazardous control behavior has been identified, design features are used to eliminate or control it or, if the system design already exists, the design is analyzed to determine if the potentially hazardous behavior has been eliminated or controlled in that design. Accomplishing this goal requires more information about the cause of the behavior, and this information is identified using the second step of STPA. The control structure diagram is evaluated using the potential control flaws in Fig. 2.

As an example of further analysis in this step, the causal factors that can result in violating the safety constraint SC1.1 are shown in Fig. 11. In the figure, t and $\boldsymbol{x}$ denote the time elapsed since the HTV is deactivated and the HTV's state vector, respectively. As required by the HTV flight rules, the ISS crew must capture the HTV within 99 s from deactivation; otherwise, the HTV must be activated again. In addition, if the ISS crew confirms by the state vector feedback or visual monitoring that the HTV drifts out of the capture box, the HTV must be activated again. Therefore, t, $\boldsymbol{x}$, the HTV flight mode (activated or deactivated), and visual information are the critical information for the crew to make an appropriate decision. If any of them is missing or inadequate, the crew must send an activation command to the HTV.

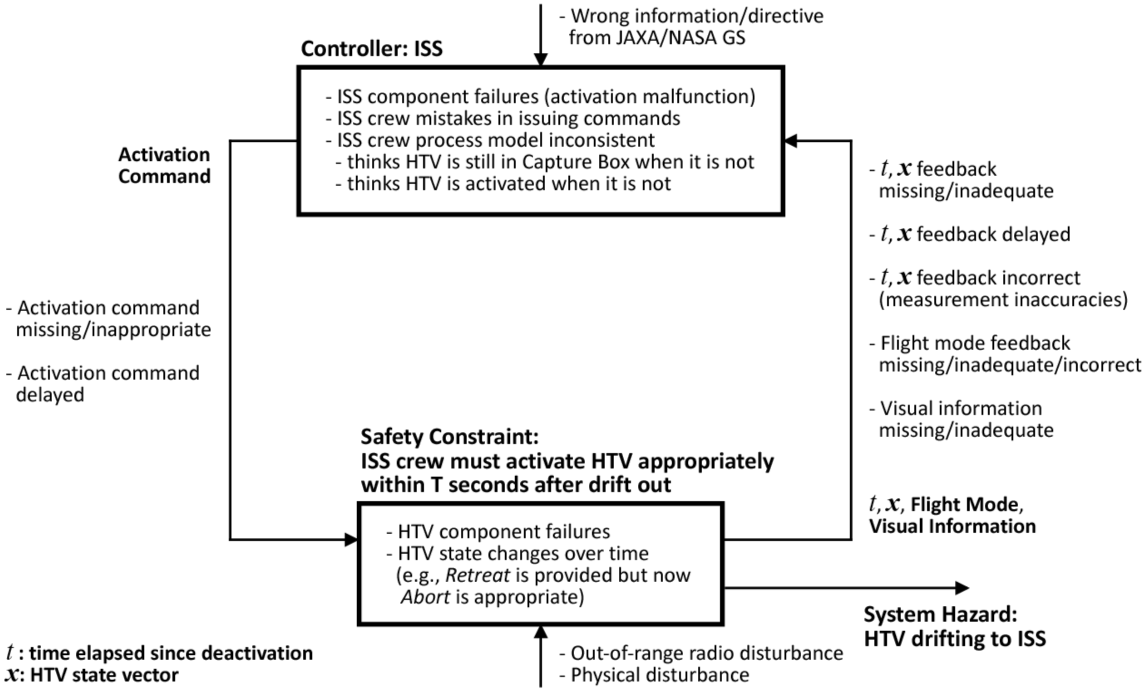
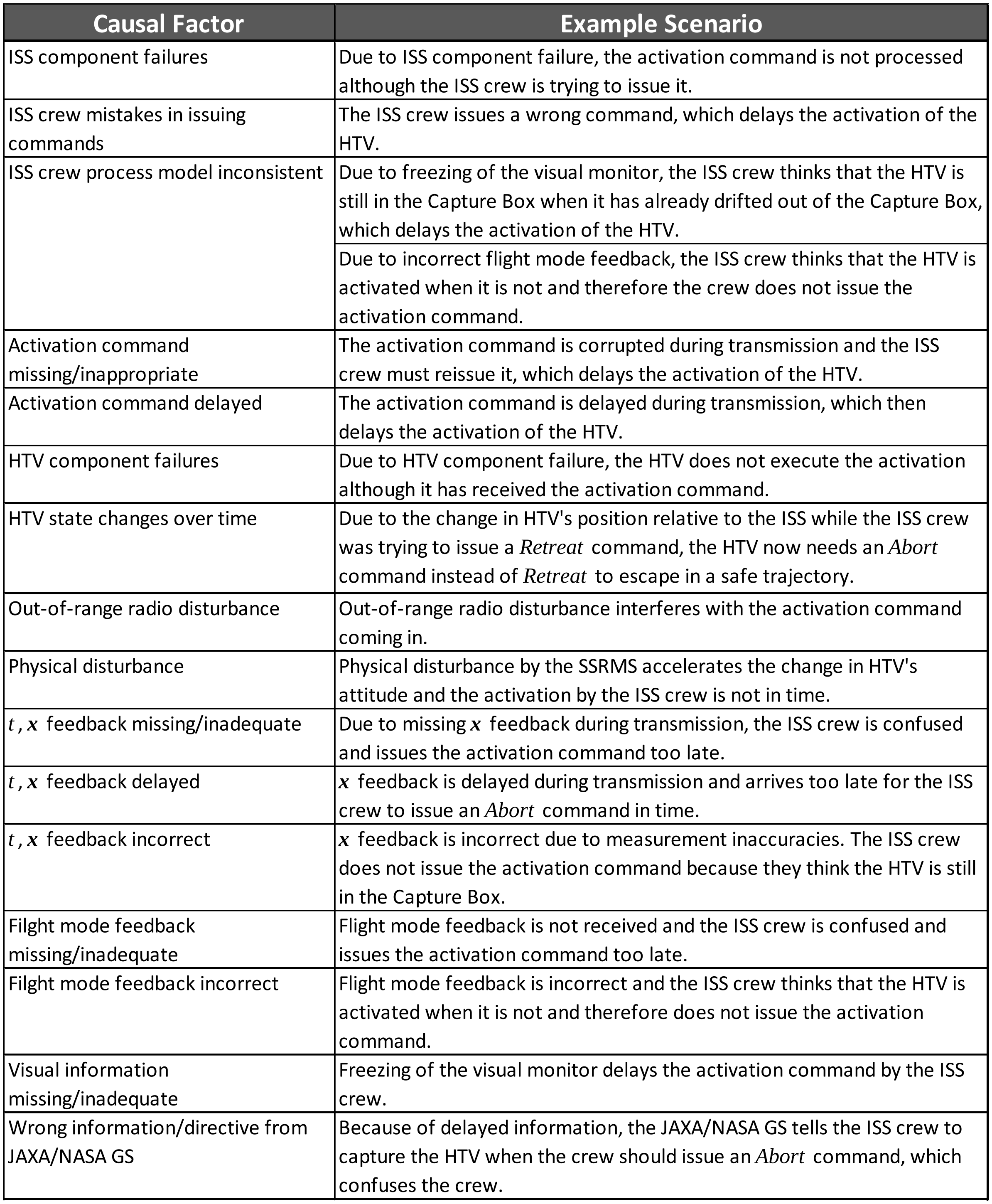

- Wrong information/directive from JAXA/NASA GS
Controller: ISS
- ISS component failures (activation malfunction)
- ISS crew mistakes in issuing commands
- ISS crew process model inconsistent
- thinks HTV is still in Capture Box when it is not
- thinks HTV is activated when it is not
Activation Command
- Activation command missing/inappropriate
- Activation command delayed
- t, x feedback missing/inadequate
- t, x feedback delayed
- t, x feedback incorrect (measurement inaccuracies)
- Flight mode feedback missing/inadequate/incorrect
- Visual information missing/inadequate
Safety Constraint:
ISS crew must activate HTV appropriately within T seconds after drift out
- HTV component failures
- HTV state changes over time (e.g., Retreat is provided but now Abort is appropriate)
t, x, Flight Mode, Visual Information
System Hazard: HTV drifting to ISS
- Out-of-range radio disturbance
- Physical disturbance
t : time elapsed since deactivation
x: HTV state vector

Fig. 11 Causal factors violating safety constraint SC1.1.

Fig. 12 shows examples of hazardous scenarios that could violate the safety constraint SC1.1. All the factors could lead to no activation or a late activation after drift out of the capture box, which would contribute to a collision with the ISS. One of the causes considered by STPA is crew process model inconsistency. For example, if the HTV was designed such that the flight mode feedback could be returned prematurely before it really was activated, an inconsistency could result. This kind of hazard cause should ideally be identified in early development and eliminated by the design. If the system already exists, as in the case of the HTV, the design must be evaluated with respect to each of these potential causal factors to determine whether the design prevents it or whether preventive or mitigation measures must be added.



Causal Factor	Example Scenario
ISS component failures	Due to ISS component failure, the activation command is not processed although the ISS crew is trying to issue it.
ISS crew mistakes in issuing commands	The ISS crew issues a wrong command, which delays the activation of the HTV.
ISS crew process model inconsistent	Due to freezing of the visual monitor, the ISS crew thinks that the HTV is still in the Capture Box when it has already drifted out of the Capture Box, which delays the activation of the HTV.
	Due to incorrect flight mode feedback, the ISS crew thinks that the HTV is activated when it is not and therefore the crew does not issue the activation command.
Activation command missing/inappropriate	The activation command is corrupted during transmission and the ISS crew must reissue it, which delays the activation of the HTV.
Activation command delayed	The activation command is delayed during transmission, which then delays the activation of the HTV.
HTV component failures	Due to HTV component failure, the HTV does not execute the activation although it has received the activation command.
HTV state changes over time	Due to the change in HTV's position relative to the ISS while the ISS crew was trying to issue a *Retreat* command, the HTV now needs an *Abort* command instead of *Retreat* to escape in a safe trajectory.
Out-of-range radio disturbance	Out-of-range radio disturbance interferes with the activation command coming in.
Physical disturbance	Physical disturbance by the SSRMS accelerates the change in HTV's attitude and the activation by the ISS crew is not in time.
$t, \boldsymbol{x}$ feedback missing/inadequate	Due to missing $\boldsymbol{x}$ feedback during transmission, the ISS crew is confused and issues the activation command too late.
$t, \boldsymbol{x}$ feedback delayed	$\boldsymbol{x}$ feedback is delayed during transmission and arrives too late for the ISS crew to issue an *Abort* command in time.
$t, \boldsymbol{x}$ feedback incorrect	$\boldsymbol{x}$ feedback is incorrect due to measurement inaccuracies. The ISS crew does not issue the activation command because they think the HTV is still in the Capture Box.
Filght mode feedback missing/inadequate	Flight mode feedback is not received and the ISS crew is confused and issues the activation command too late.
Filght mode feedback incorrect	Flight mode feedback is incorrect and the ISS crew thinks that the HTV is activated when it is not and therefore does not issue the activation command.
Visual information missing/inadequate	Freezing of the visual monitor delays the activation command by the ISS crew.
Wrong information/directive from JAXA/NASA GS	Because of delayed information, the JAXA/NASA GS tells the ISS crew to capture the HTV when the crew should issue an *Abort* command, which confuses the crew.

Fig. 12 Example scenarios violating safety constraint SC1.1.

3. Analysis of the Results of the Case Study

The feasibility of applying STPA to the HTV was demonstrated by the above analysis. In order to determine the usefulness of STPA, the hazard causes identified by STPA were compared with the existing FTA results. The objective of this comparison task was to answer the following two questions:

1) Do the hazard causes identified by STPA include the causes identified by FTA?

2) Did STPA find additional causes that had not been identified by FTA?

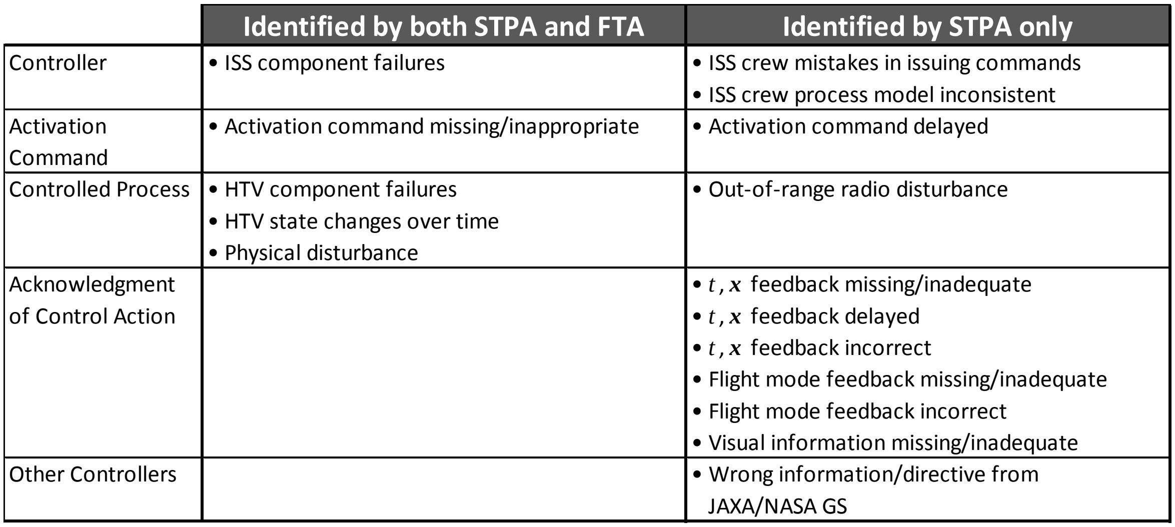
In the comparison, the fault tree branches for the capture operation (previously prepared for the HTV) were compared with the STPA results by mapping the STPA hazard causes to the fault tree branches to identify the differences.

The results from the comparison analysis answered the above two questions as follows:

1) All the causal factors identified by FTA were found by STPA.

2) STPA identified additional causal factors that had not been identified by FTA.

Causal factors that were identified by both methods and those by STPA only are listed in Fig.13. There were no causal factors identified by FTA that were not identified by STPA. Note that the people performing STPA were not familiar with the existing fault tree analysis and thus were not biased by them.



	Identified by both STPA and FTA	Identified by STPA only
Controller	• ISS component failures	• ISS crew mistakes in issuing commands • ISS crew process model inconsistent
Activation Command	• Activation command missing/inappropriate	• Activation command delayed
Controlled Process	• HTV component failures • HTV state changes over time • Physical disturbance	• Out-of-range radio disturbance
Acknowledgment of Control Action		• t, x feedback missing/inadequate • t, x feedback delayed • t, x feedback incorrect • Flight mode feedback missing/inadequate • Flight mode feedback incorrect • Visual information missing/inadequate
Other Controllers		• Wrong information/directive from JAXA/NASA GS

Fig. 13 General causal factors identified by both FTA and STPA and by STPA only.

The result is not surprising as FTA concentrates on component failure while STPA considers such failures as well as other types of unsafe control such as process model inconsistency and causal factors related to delay of commands, delay of feedback, and acknowledgment of control actions. Some causal factors identified by the STPA are due to control flaws in the control loop involving total system integration among the ISS, HTV, and NASA/JAXA GS. In contrast, most of basic events identified by the FTA were events that occur by chance, such as component failures, and not potential flaws in the basic system design.

In the review of the HTV design after the STPA analysis, some of the additional causal factors were found to have been considered and controlled in the actual HTV design and operation but were not explicitly identified by the FTA.

One example is, of course, not proof of anything. The evidence on real systems, however, in a variety of industries is accumulating. A few examples follow. The Missile Defense Agency tried STPA for a predeployment nonadvocate safety assessment of the new U.S. Ballistic Missile Defense System for the hazard of inadvertent launch [8]. The system had been subjected to the traditional hazard analysis methods used in the defense sector. STPA was performed by 2 people over 6 months who started with no familiarity with the system. So many previously unknown scenarios for inadvertent launch were found that deployment and testing were delayed for 6 months. In many of these scenarios, all the components were operating exactly as intended, but the complexity of the component interactions led to unanticipated system behavior. Examples include missing cases in software requirements and timing problems in sending and receiving messages. STPA also identified component failures that could cause the hazard.

The analysis of a new Air Traffic Control procedure (performed by two students) found more hazardous scenarios than a team of experts had found [9]. In the analysis of a blood gas analyzer, STPA found 175 scenarios versus 75 found by a FMEA. The STPA analysis took much less time and resources (one person for a few weeks versus the FMEA, which required a team over many months). Only the STPA found the scenario that had led to the near death of a patient and a recall of the device by the FDA [10]. In fact, nine scenarios were found by the STPA that could lead to the hazardous behavior. A recent comparative study in nuclear power plants compared several different techniques, including STPA, fault trees, and FMEA. Only STPA found the scenario that had actually led to an accident in that plant. None of the analysts, of course, knew beforehand about the accident [11].

C. Identifying Potentially Unsafe Interactions among Multiple Controllers

A second cause of accidents (see Sec. II) is potential interference among uncoordinated control actions by multiple controllers. The collision between two aircraft over Überlingen, Germany, has been partially blamed on conflicting advisories to the pilots by the ground air traffic controller and TCAS II, the automated collision avoidance system on the two aircraft [13]. One aircraft followed the ground advisory while the other aircraft followed the TCAS advisory. Other types of conflicts between multiple controllers are possible.

In the previous section, a process for identifying unsafe control actions was described. Individually safe control actions by different controllers over common components, however, can interact in such way as to lead to a hazard. This section describes a procedure for identifying such hazardous scenarios.

Fig. 14 is a variant of the table used in Step 1 STPA (Fig. 9) modified to consider multiple controllers. For interactions among N (>2) controllers, this tabular form can be extended to N dimensions. Each cell in the table represents one of the following four types of interactions:

[A] denotes that only one safe control action is provided. This case does not lead to a hazard and need not be analyzed further.

[B] denotes that multiple “individually safe” control actions are provided. This case could lead to a hazard. As one example, each controller may provide the same “safe” command to the system, with the second command overriding the second one, starting the process over and potentially exceeding the time limits for action. For these situations, the system should be designed or controlled such that only one safe control action is executed, even if a (possibly redundant) additional safe command is provided.

[C] denotes that both “individually safe” and unsafe control actions are provided. For these potentially unsafe scenarios, the system needs to be designed or controlled such that the safe control action is properly executed without being interrupted by unsafe ones.

[D] denotes that only unsafe control actions are provided. Designing the proper response to this case requires further causal analysis to identify the detailed scenarios that can be involved.

	1	Control Action by Controller 1		
2		"Individually Safe" Causes Hazard	Not Providing Causes Hazard	Providing Causes Hazard
Control Action by Controller 2	"Individually Safe" Causes Hazard	[B] Multiple "individually safe" control actions are provided	[A] Only one safe control action is provided	[C] Both "individually safe" and unsafe control actions are provided
	Not Providing Causes Hazard	[A] Only one safe control action is provided	[D] Only unsafe control actions are provided	[D] Only unsafe control actions are provided
	Providing Causes Hazard	[C] Both "individually safe" and unsafe control actions are provided	[D] Only unsafe control actions are provided	[D] Only unsafe control actions are provided

Fig. 14 Potentially unsafe interactions of control actions between two controllers.

To identify causal scenarios for multiple unsafe control actions, the entire system must be considered from each controller's point of view. Fig. 15 is similar to Fig. 2 but with multiple controllers of a common process. One difference is that each controller must also have a model of the other controller as well as a model of the controlled process.

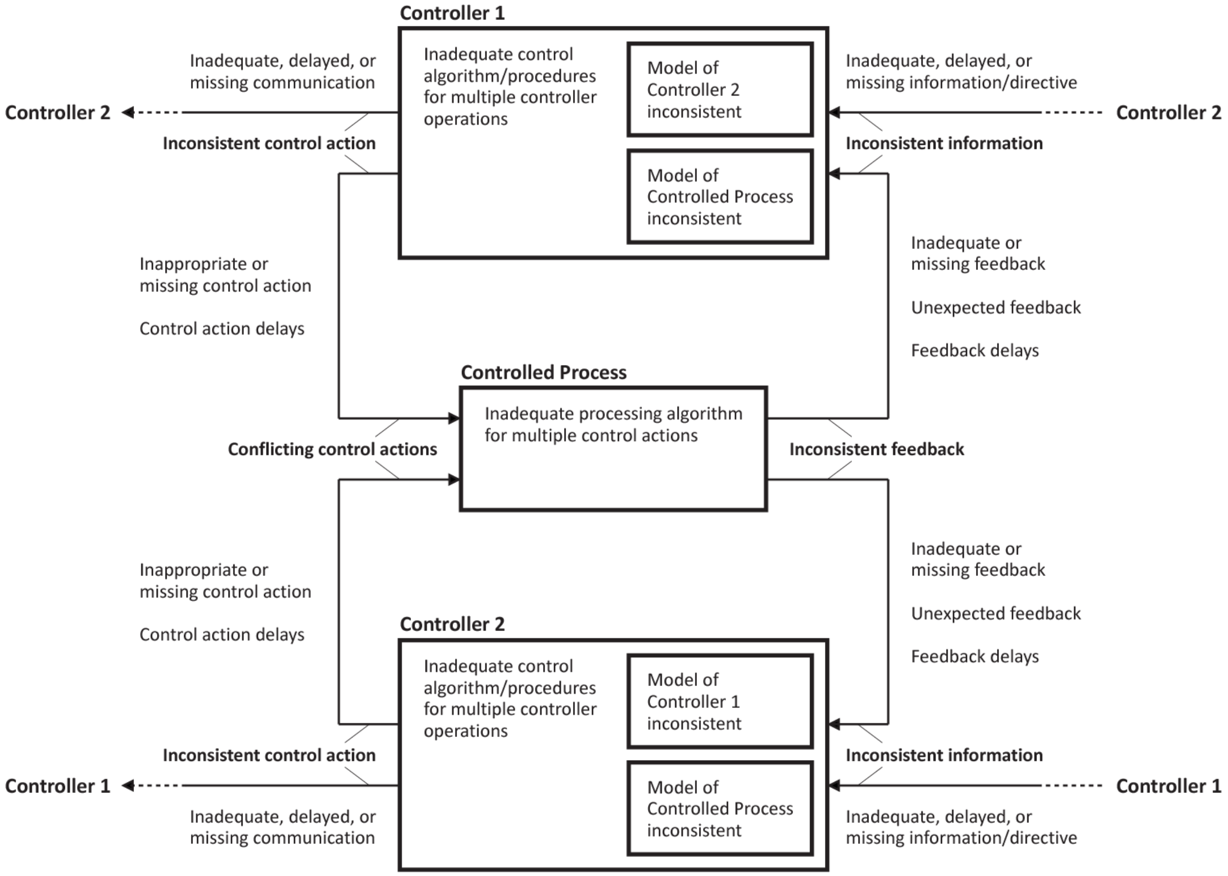
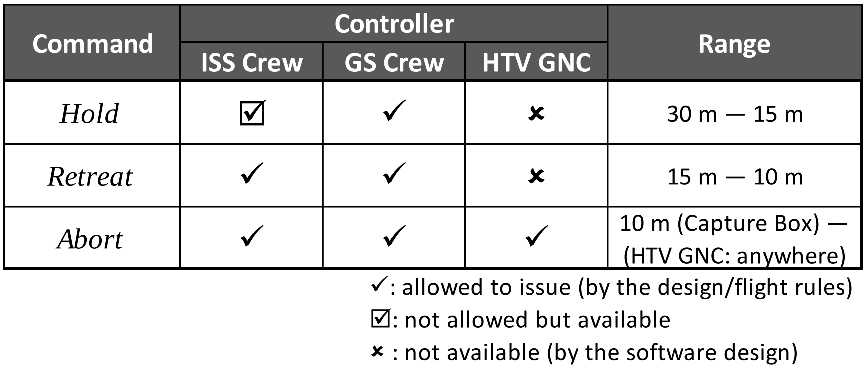

Controller 1
Inadequate control algorithm/procedures for multiple controller operations
Model of Controller 2 inconsistent
Model of Controlled Process inconsistent
Inadequate, delayed, or missing communication
Controller 2
Inconsistent control action
Inadequate, delayed, or missing information/directive
Controller 2
Inconsistent information
Inappropriate or missing control action
Control action delays
Inadequate or missing feedback
Unexpected feedback
Feedback delays
Controlled Process
Inadequate processing algorithm for multiple control actions
Conflicting control actions
Inconsistent feedback
Inappropriate or missing control action
Control action delays
Inadequate or missing feedback
Unexpected feedback
Feedback delays
Controller 2
Inadequate control algorithm/procedures for multiple controller operations
Model of Controller 1 inconsistent
Model of Controlled Process inconsistent
Inconsistent control action
Controller 1
Inadequate, delayed, or missing communication
Inconsistent information
Controller 1
Inadequate, delayed, or missing information/directive

Fig. 15 Causal factors leading to unsafe interaction between double controllers.

After the causal factors have been identified for each individual controller using Fig. 15 as a guide, specific hazardous scenarios can be built by combining the causal factors from each controller. For each possible hazardous scenario identified, each causal factor leading to this scenario should be eliminated or controlled in the system design or in operations. The HTV final approach phase is used as an example.

The first case study focused on the HTV capture operation at the capture box. This second case study focused on the final approach phase from 30 m below the ISS up to the capture box. In this phase, the HTV performs an automatic approach to the ISS without any commands by the ISS crew or the ground station (GS) crew in the nominal case. If an emergency occurs, the ISS and GS crew can send off-nominal commands such as *Hold*, *Retreat*, and *Abort*. In addition to these two controllers, the HTV itself is capable of executing an *Abort*. Thus, this phase can be viewed as a triple-controller situation.

Fig. 16 shows the availability and range of the off-nominal commands for each of the three controllers. In case of an emergency, the ISS and GS crew are supposed to issue *Hold*, *Retreat*, and *Abort* in the ranges of 30 to 15 m, 15 to 10 m, and the capture box and beyond, respectively, while the HTV itself can execute an abort anywhere. If any of these commands are not provided, the HTV could eventually collide with the ISS. In other words, *Abort* is obviously the most critical command to avoid the collision because it is the final line of defence before the HTV collides with the ISS. For this reason, the *Abort* command is the focus of the example analysis.



Command	Controller			Range
	ISS Crew	GS Crew	HTV GNC	
Hold	☑	✓	✗	30 m — 15 m
Retreat	✓	✓	✗	15 m — 10 m
Abort	✓	✓	✓	10 m (Capture Box) — (HTV GNC: anywhere)

✓: allowed to issue (by the design/flight rules)
☑: not allowed but available
✗ : not available (by the software design)

Fig. 16 Off-nominal command availability and range.

Fig. 17 shows a detailed control structure for the HTV final approach phase, each controller augmented with a model of the controlled process and a model of the other controllers. The control structure for this phase, again, is composed of three controllers: the ISS crew, the GS crew, and the HTV Guidance Navigation and Control (GNC). For simplicity, the GS here represents both NASA and JAXA ground stations. The Tracking and Data Relay Satellite (TDRS) as a backup communication system is omitted to simplify the diagram. There is a voice loop connection between the ISS crew and the GS crew so that they can communicate with each other through the entire operation. The three off-nominal commands, *Hold*, *Retreat*, and *Abort*, as well as *Free Drift* are identified in Fig. 17 as control actions provided by the ISS and GS crew.

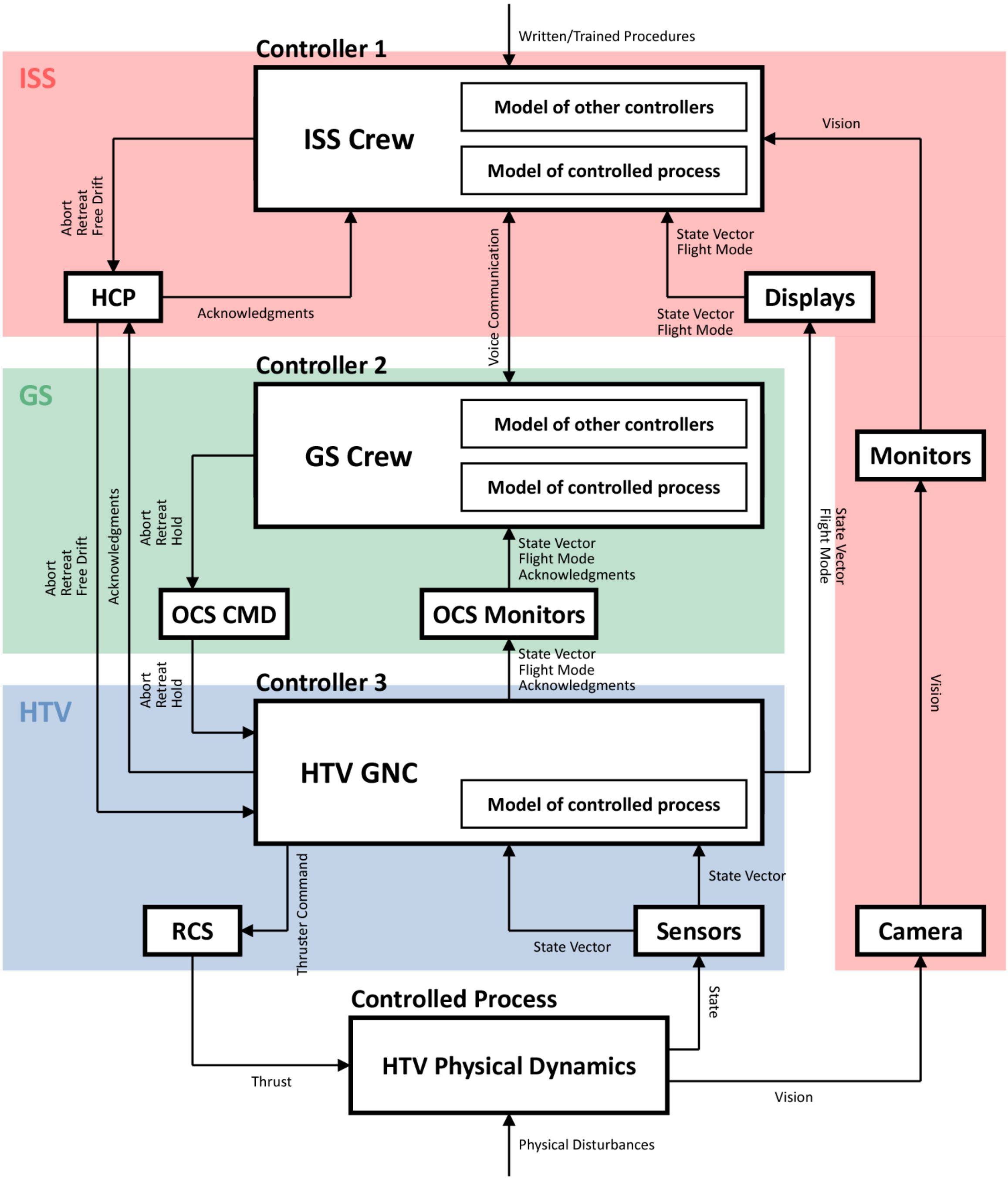


Fig. 17 Control structure for HTV final approach phase.

The first step of the analysis identifies unsafe interactions of multiple control actions using a table like the table shown in Fig. 14. Because there are three controllers in this case, a three-dimensional table is needed. The case considered is one where the HTV must be aborted immediately.

Instead of *Abort*, the ISS crew could incorrectly provide *Retreat* or *Free Drift* while the GS crew could incorrectly provide *Retreat* or *Hold*. The HTV GNC is allowed to issue only the *Abort* command. Therefore, the following unsafe control actions for each controller must be considered:

ISS Crew: Individually safe, not providing, and providing *Retreat* or *Free Drift*
GS Crew: Individually safe, not providing, and providing *Retreat* or *Hold*
HTV GNC: Individually safe and not providing *Abort*

Fig. 18 and Fig. 19 show the hazardous interactions identified. Each cell in the table is classified as one of the [A], [B], [C] and [D] categories already described.

Detailed scenarios leading to these unsafe combinations of commands can be identified using combinations of the causal factors shown in Fig. 15. As an example, one possible scenario leading to D2 is 1) the ISS crew issues a *Retreat* command before the HTV initiates a self-abort because they do not want to waste time and fuel by starting all the final approach process over again; 2) the GS crew is satisfied with the *Retreat* provided by the ISS crew and no longer pays close attention; and 3) because a *Retreat* command has been provided by the ISS crew, the HTV GNC does not self-abort.

Not all the causal factors leading to an unsafe interaction arise from multiple controller contributions: normal causal factors that are not related to multiple controllers, such as component failures and other causes as listed in Fig. 2, could also be a trigger of unsafe interactions among multiple controllers. Therefore, the multiple controller hazards cannot be considered independently.

Identifying potentially hazardous combinations of multiple control actions may be adequate to design prevention and mitigation measures. More detailed causal analysis will allow more effective and targeted mitigation measures.

HTV GNC *Abort* : "Individually Safe" Causes Hazard

GS \ ISS			ISS Crew *Abort*			
			"Individually Safe" Causes Hazard	Not Providing Causes Hazard	Providing Causes Hazard	
					Retreat	*Free Drift*
GS Crew *Abort*	"Individually Safe" Causes Hazard		[B1] Triple *Abort* commands are redundantly provided by ISS crew, GS crew, and HTV GNC	[B2] Double *Abort* commands are redundantly provided by GS crew and HTV GNC	[C1] Double *Abort* commands are provided by GS crew and HTV GNC while a *Retreat* command is provided by ISS crew	[C2] Double *Abort* commands are provided by GS crew and HTV GNC while a *Free Drift* command is provided by ISS crew
	Not Providing Causes Hazard		[B3] Double *Abort* commands are redundantly provided by ISS crew and HTV GNC	[A1] Only a single *Abort* command is provided by HTV GNC	[C3] An *Abort* command is provided by HTV GCN while a *Retreat* command is provided by ISS crew	[C4] An *Abort* command is provided by HTV GNC while a *Free Drift* command is provided by ISS crew
	Providing Causes Hazard	*Retreat*	[C5] Double *Abort* commands are provided by ISS crew and HTV GNC while a *Retreat* command is provided by GS crew	[C6] An *Abort* command is provided by HTV GNC while a *Retreat* command is provided by GS crew	[C7] An *Abort* command is provided by HTV GNC while double *Retreat* commands are provided by ISS crew and GS crew	[C8] An *Abort* command is provided by HTV GNC while a *Retreat* command is provided by GS crew and a *Free Drift* command is provided by ISS crew
		Hold	[C9] Double *Abort* commands are provided by ISS crew and HTV GNC while a *Hold* command is provided by GS crew	[C10] An *Abort* command is provided by HTV GNC while a *Hold* command is provided by GS crew	[C11] An *Abort* command is provided by HTV GNC while a *Retreat* command is provided by ISS crew and a *Hold* command is provided by GS crew	[C12] An *Abort* command is provided by HTV GNC while a *Free Drift* command is provided by ISS crew and a *Hold* command is provided by GS crew

Fig. 18 Unsafe interactions between the three controllers (HTV GNC *Abort*: "Individually Safe" Causes Hazard).

					HTV GNC *Abort* : Not Providing Causes Hazard	
GS		ISS	ISS Crew *Abort*			
			"Individually Safe" Causes Hazard	Not Providing Causes Hazard	Providing Causes Hazard	
					Retreat	*Free Drift*
GS Crew *Abort*	"Individually Safe" Causes Hazard		[B4] Double *Abort* commands are redundantly provided by ISS crew and GS crew	[A2] Only a single *Abort* command is provided by GS crew	[C13] An *Abort* command is provided by GS crew while a *Retreat* command is provided by ISS crew	[C14] An *Abort* command is provided by GS crew while a *Free Drift* command is provided by ISS crew
	Not Providing Causes Hazard		[A3] Only a single *Abort* command is provided by ISS crew	[D1] No *Abort* command is provided by any of the three controllers	[D2] Only a *Retreat* command is provided by ISS crew	[D3] Only a *Free Drift* command is provided by ISS crew
	Providing Causes Hazard	*Retreat*	[C15] An *Abort* command is provided by ISS crew while a *Retreat* command is provided by GS crew	[D4] Only a *Retreat* command is provided by GS crew	[D5] Double *Retreat* commands are provided by ISS crew and GS crew	[D6] A *Free Drift* command is provided by ISS crew and a *Retreat* command is provided by GS crew
		Hold	[C16] An *Abort* command is provided by ISS crew while a *Holde* command is provided by GS crew	[D7] Only a *Hold* command is provided by GS crew	[D8] A *Retreat* command is provided ISS crew and a *Hold* command is provided by GS crew	[D9] A *Free Drift* command is provided by ISS crew and a *Hold* command is provided by GS crew

Fig. 19 Unsafe interactions between the three controllers (HTV GNC *Abort*: Not Providing Causes Hazard).

Up to this point, unsafe interactions have been considered by looking at the combinations of control actions only. However, the context of each control action is also a key factor involved in unsafe interactions. A table like that shown in Fig. 14 is useful to identify the combinations, but context is important because multiple control actions can have differing temporal relationships. Some contexts are unsafe while others are not, even if they are composed of the same control actions. One type of analysis of the potential scenarios involves identifying the unsafe contexts.

Fig. 20 shows an example. In this diagram, time is shown horizontally and the controllers are shown vertically. The changes of the state of the controlled process are denoted with color and fill patterns. . In this context, there are two states, the actual state and the state expected by the controller. These two states are not always consistent. A controller expects that a controlled process is changed from a specific state to another by its control action. This is represented in Fig. 20 by a diagonally divided control action with color and fill patterns (e.g. "Issue Control Action 1", "Issue Control Action 2"). However the actual state of the controlled process could be changed by the other controller without the first controller knowing it. As a result, the first controller could execute an inconsistent control action with the actual state of the controlled process, such as "Issue Control Action 1" in Fig. 20.

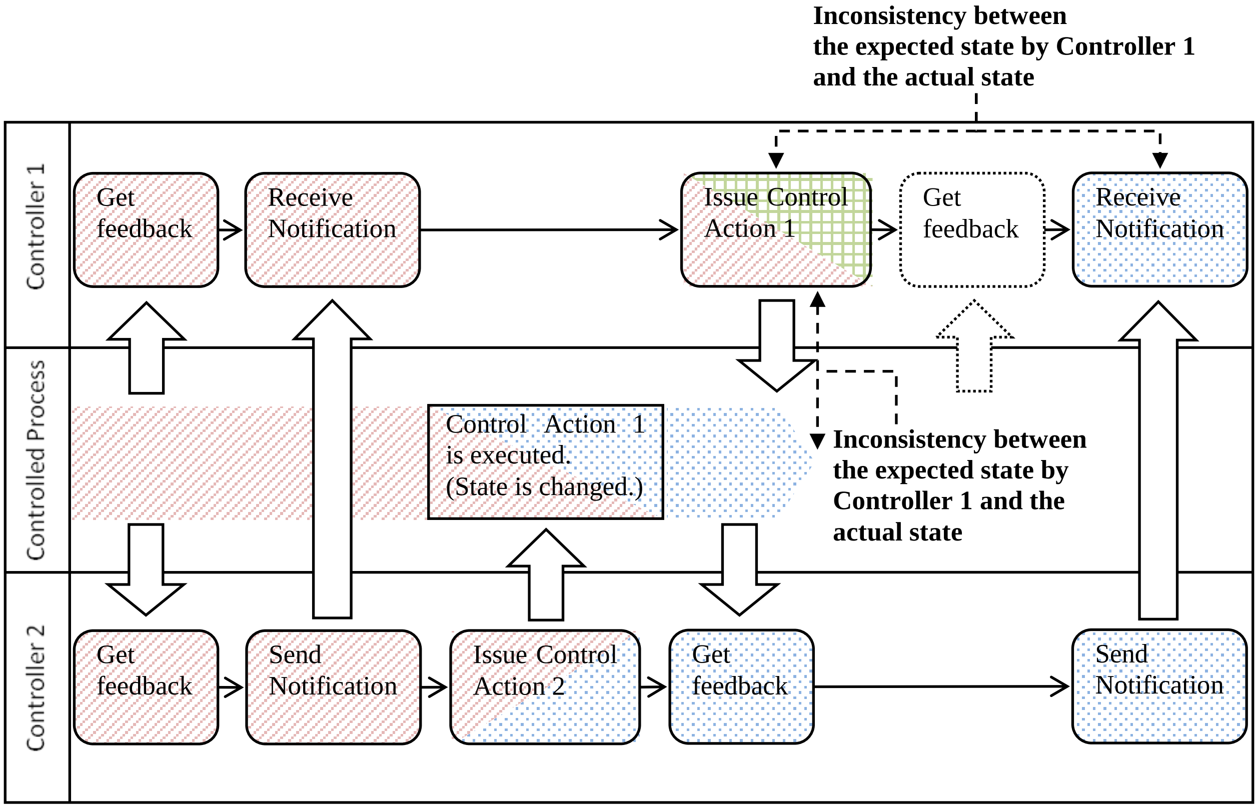


Fig. 20 Example of context diagram

In Fig. 20, Controller 1 and Controller 2 first receives feedback about the state of the controlled process. Controller 2 then sends a notification to Controller 1. Controller 2 issues Control Action 2 and the state of the Controlled Process is changed. Next, Controller 2 gets feedback from the controlled process. Controller 1 issues Control Action 1 without knowing that the controlled process state has been changed. Controller 2 sends notification to Controller 1 but Controller 1 has already sent Control Action 1 based on the previous state of the process.

Although context can be analyzed with a context diagram, it would be inefficient and unrealistic to generate all possible contexts. Instead, preconditions and postconditions of the control actions can be used to define and identify interference. The preconditions can involve the state of controller itself, the state of controlled process, external conditions of system, and so on. Postconditions denote changes to the state of the controlled process and the controller itself as a result of the control action. Interference among control actions occurs when 1) the preconditions of control actions are inconsistent or 2) the postcondition of a control action is inconsistent with a precondition of another control action.

	Precondition	**Postcondition**
Control Action 1	・Receiving feedback ・Receiving Notification	・Updating feedback (Changing the state of Controlled Process) ・Issuing Notification
Control Action 2	・Receiving feedback	・Updating feedback (Changing the state of Controlled Process) ・Issuing Notification

Fig. 21 Example condition table for Fig. 20

Fig. 21 shows the preconditions and postconditions for the example in Fig. 20. Obviously, the postcondition "Issuing Notification" (shown in red) of Control Action 2, can change the contents of one of the preconditions of Control Action 1, “Receiving Notification" shown in red.

. This approach to identifying interference can be applied not only to nominal control actions but also potentially unsafe control actions in each single controller (“not providing causes hazard,” “providing causes hazard,” “wrong timing/order causes hazard,” and “stopping too soon/applying too long causes hazard.”).

SpecTRM, a system and software engineering environment that supports safety engineering processes such as hazard analysis [13] was used to experimentally validate the feasibility of using this method to identify interference. SpecTRM allows modeling preconditions and postconditions and automated checking for consistency and interference. Focusing on the "B type" combinations between GS (ground station) crew and ISS crew, 15 types (preconditions) of individually safe control actions by the JAXA GS crew and 8 by the ISS Crew were analyzed. Several unsafe interferences between the ISS Crew and the GS crew were identified even when only individually safe control actions are issued.

Fig. 22 and Fig. 23 show the preconditions and postconditions for the GS Crew and ISS Crew to execute Abort and Hold. In the figures, the conditions about the flight mode (FM) of the HTV, the HTV's position (state vector or SV), the state of a sensor (Rendezvous Sensor State or RVSS), are specified mathematically. SV3, SV4 and SV5 are distance data of the HTV, with each one indicating a different kind of distance. "SV3 > 70", for example, means the distance is over 70 m. Fig. 24 shows the context diagram. In this case, the ISS crew issues a *Hold* command after the GS crew issues an *Abor*t. This control sequence is potentially hazardous because the postcondition of the Abort (FM = Abort) is inconsistent with one of the preconditions of the Hold (FM = Approach **OR** FM = Retreat). Because it is hazardous for the HTV to stop temporarily or not to finish after it starts an abort, the safety constraint that the HTV must finish aborting once it starts to abort must imposed on the HTV. To avoid this hazard, the actual HTV system is designed to reject a *Hold* command by the HTV GNC when the HTV is aborting.

No	Control Action	**Precondition**	**Postcondition**
1	Hold 1	(FM = Approach OR FM = Hold OR FM = Retreat) AND ((SV3 > 70 AND 200 < SV4 < 250) OR (SV3 > 40 AND 100 < SV4 < 200) OR (SV3 > 15 AND 30 < SV4 < 100) OR (SV3 > 5 AND 15 < SV4 < 30) OR (SV3 > 3.7 AND 15 < SV4 < CAPTURE_POINT))	FM = Abort

Fig. 22 Part of a condition table of GS Crew.

No	Control Action	**Precondition**	**Postcondition**
1	Hold 1	30 < SV4 < 250 AND VC = broken AND SV1 < KOS AND RVSS = ONLY ONE FUNCTIONING AND SV5 > 15 AND (FM = Approach OR FM = Retreat)	FM = Hold

Fig. 23 Part of a condition table of ISS Crew.

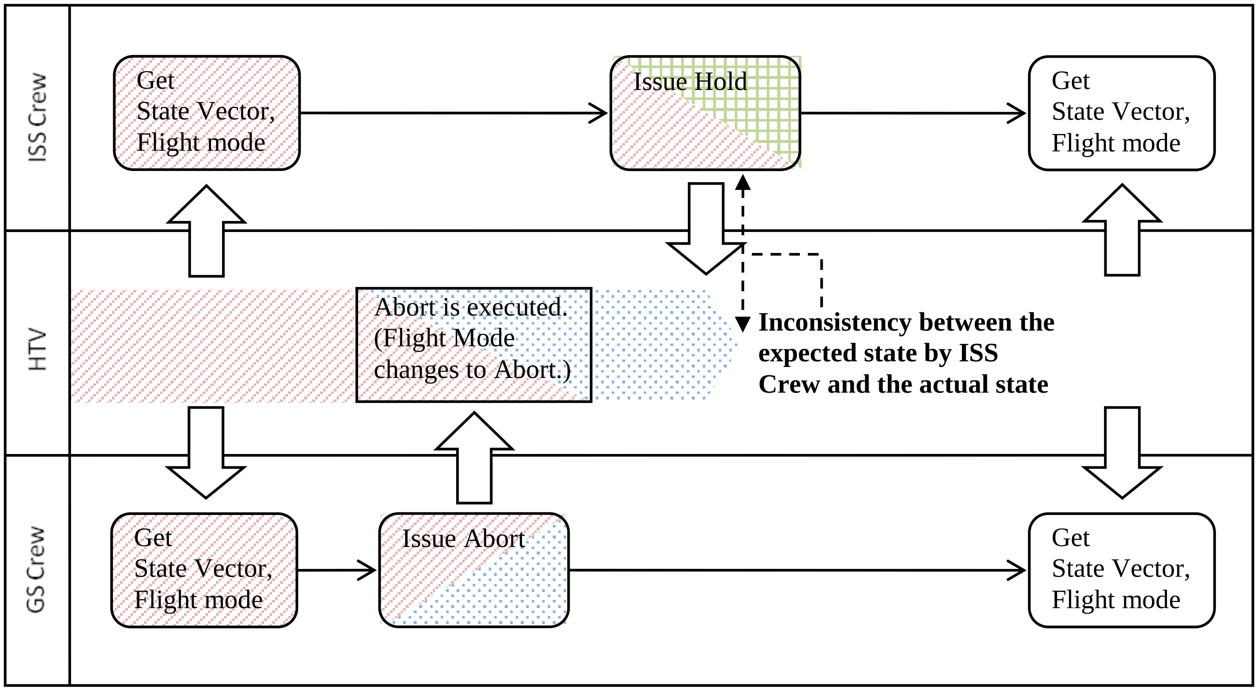

ISS Crew
HTV
GS Crew
Get State Vector, Flight mode
Issue Hold
Get State Vector, Flight mode
Abort is executed. (Flight Mode changes to Abort.)
Inconsistency between the expected state by ISS Crew and the actual state
Get State Vector, Flight mode
Issue Abort
Get State Vector, Flight mode

Fig. 24 Example of context diagram

IV. Conclusions

As spacecraft become more and more complex, the limitations of traditional hazard analysis techniques are revealed. More powerful techniques are needed that can handle the new causes of accidents in these systems. This paper presents a new hazard analysis technique called STPA (System-Theoretic Process Analysis). STPA was illustrated on the analysis of the Japanese Aerospace Exploration Agency (JAXA) H-II Transfer Vehicle capture and approach phases.

Because a fault tree analysis had been performed on the H-II Transfer Vehicle, a comparison of the results of the two analysis techniques was possible. For the first case study, STPA identified a total of 22 unsafe control actions and seven hazards for the capture operation. One of the hazards (and one of its safety constraints) was selected for further analysis and the STPA results compared with the existing fault tree analysis-based hazard report for the spacecraft. The comparison showed that STPA identified the causal factors identified in the fault tree analysis, but STPA also identified additional causal factors that had not been identified by fault tree analysis. The additional factors include those that *cannot* be identified using fault tree analysis, including software and system design as well as system integration of the International Space Station, the H-II Transfer Vehicle, and the NASA/JAXA Ground Stations.

The second case study showed how STPA can identify possible unsafe interactions among multiple controllers that are caused by conflicting or uncoordinated control actions. The multiple controller problem cannot be captured by fault tree analysis, and, therefore, no comparison was necessary.

Finding unsafe design errors after a system is already designed is not as useful as using hazard analysis to drive the design from the beginning. Because STPA treats safety as a control problem, an existing design is not necessary to perform the analysis (as is required for fault tree analysis and failure modes and effects analysis), and potentially the analysis and design processes can proceed in parallel, each supporting the other.

Acknowledgments

This research was performed at Massachusetts Institute of Technology (MIT) and Japan Aerospace Exploration Agency (JAXA) under a contract with Japan Manned Space Systems Corporations (JAMSS). The authors would like to thank members of JAXA, JAMSS, and the Complex Systems Research Laboratory (CSRL) at MIT, and Assoc. Prof. Seiko Shirasaka at Keio University for their support in this study.

References

[1] Albee et al (JPL Special Review Board), “Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions,” NASA Jet Propulsion Laboratory, 22 March 2000.

[2] Leveson N. G., “Role of Software in Spacecraft Accidents,” *Journal of Spacecraft and Rockets*, Vol. 41, No. 4, 2004, pp. 564-575.

[3] Leveson N. G., “Software Challenges in Achieving Space Safety,” *Journal of the British Interplanetary Society*, Vol. 62, July/August 2009, pp. 265-272, doi:1721.1/58930.

[4] Leveson, N. G., *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press, January 2012.

[5] Pavlovich, J.G., *Formal Report of Investigation of the 30 April 1999 Titan IV B/Centaur TC-14/Milstar-3 (B-32) Space Launch Mishap*, U.S. Air Force, 1999.

[6] Japan Aerospace Exploration Agency, *HTV-1 Mission Press Kit*, Japanese Aerospace Exploration Agency, September 9, 2009, Tokyo Japan.

[7] Japan Aerospace Exploration Agency, *HTV 2 (KOUNOTORI 2) Mission Press Kit*, Japanese Aerospace Exploration Agency, January 20, 2011.

[8] Pereire, S., Lee, G., and Howard, J., “A System-Theoretic Hazard Analysis Methodology for a Non-Advocate Safety Assessment of the Ballistic Missile Defense System,” Proceedings of the 2006 AIAA Missile Sciences Conference Held in Monterey, California on 14-16 November 2006.

[9] Leveson, N.G., Fleming, C.H., Spencer, M., Thomas, J., Safety Assessment of Complex, Software-Intensive Systems, Proceedings of the SAE International Journal of Aerospace. 5(1):2012, doi:10.4271/2012-01-2134.

[10] Balgos, V., A Systems Theoretic Application to Design for the Safety of Medical Devices, SDM Master’s Thesis, Engineering Systems Division, MIT, Cambridge, MA USA, May 2012

[11] Torok, R. and Geddes, B., Nuclear Power STPA Example, Second MIT STAMP Workshop, March 2013.

[12] Investigation Report, Bundesstelle für Flugunfalluntersuchung, AX001-1-2/02, Braunschweig, May 2004.

[13] Weiss, A.A., Dulac, N., Chiesi, S., Daouk, M., Zipkin, D., and Leveson, N.G., “Engineering Spacecraft Mission Software Using a Model-Based and Safety-Driven Design Methodology,” *Journal of Aerospace Computing*, Information, and Communication, Vol. 3, November 2006, pp. 562-586.