

# MULTIPLE CONTROLLER CONTRIBUTIONS TO HAZARDS

Takuto Ishimatsu<sup>(1)</sup>, Nancy Leveson<sup>(1)</sup>, Cody Fleming<sup>(1)</sup>, Masa Katahira<sup>(2)</sup>, Yuko Miyamoto<sup>(2)</sup>, Haruka Nakao<sup>(3)</sup>

<sup>(1)</sup>Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA  
Email: takuto@mit.edu, leveson@mit.edu, chf44@mit.edu

<sup>(2)</sup>Japan Aerospace Exploration Agency, 2-1-1 Sengen, Tsukuba-shi, Ibaraki 305-8505, Japan  
Email: katahira.masafumi@jaxa.jp, miyamoto.yuko@jaxa.jp

<sup>(3)</sup>Japan Manned Space Systems Corporation, Urban Bldg., 1-1-26, Kawaguchi, Tsuchiura, Ibaraki 300-0033, Japan  
Email: haruka@jamss.co.jp

## ABSTRACT

One contributor to hazards in complex systems arises out of unsafe interactions among multiple controllers. The basic problem is that in complex systems, hazards can be created by interactions among components that are each operating “correctly.” STPA is a new hazard analysis that includes both system hazards caused by component failures (as do the traditional analysis techniques) and also those caused by unsafe interactions among components that may not have individually failed. The first descriptions of STPA, however, did not include examples of how to handle potential problems that occur between multiple controllers. We have created an approach to identify possible unsafe interactions among multiple controllers so that the system can be designed to eliminate any ambiguity or potential for unsafe controller interactions. In this paper, we describe the analysis technique and demonstrate its use for the HTV during the critical approach phase. Once these hazardous interactions are identified, they can then be eliminated or controlled through system design or operational procedures.

## 1. INTRODUCTION

One contributor to hazards in complex systems arises out of unsafe interactions among multiple controllers. For example, the 2002 Überlingen aircraft collision resulted from conflicting instructions provided by the ground Air Traffic Controller (ATC) and the onboard Traffic Collision Avoidance System (TCAS) to the crews. If both aircraft had followed the instructions from one or the other controller, the collision would have been avoided. But one crew followed the ATC instructions and the other crew followed the TCAS commands. As a result, tragedy occurred. Another example is when multiple controllers think that someone else has provided a critical command and nobody does. The usual approach to making systems safer, adding redundancy, only makes things worse in this case.

The basic problem is that in complex systems, hazards can be created by interactions among components that are each operating “correctly.” STPA is a new hazard analysis that includes both system hazards caused by component failures (as do the traditional analysis techniques) and also those caused by unsafe interactions among components that may not have individually failed. At IAASS 2010, we presented a paper that demonstrated the applications of STPA on the JAXA H-II Transfer Vehicle (HTV), an unmanned resupply spacecraft to the International Space Station (ISS), and compared the results to the classic fault tree analysis (FTA) performed on the same spacecraft [1].

The analysis technique in that paper, however, did not handle the potential problems that occur between multiple controllers. For example, critical commands such as *Abort* and *Retreat* in the approach phase may be provided for the HTV by the astronauts, by the ground controllers (NASA and JAXA), and by the software on the HTV itself. There are good reasons to allow multiple controllers to issue critical commands, but care needs to be taken that hazards do not result. We have created a technique to identify possible unsafe interactions among multiple controllers so that the system can be designed to eliminate any ambiguity or potential for unsafe controller interactions. In this paper, we describe the analysis technique and demonstrate its use for the HTV during its final approach phase.

## 2. STPA FOR MULTIPLE CONTROLLER PROBLEMS

### 2.1. Overview of STPA

Traditional hazard analysis techniques start from a mature design in order to be effective and assume that accidents are caused by component failures. Two major problems are:

- At late stages of development, the best that can be done to achieve safety is the addition of extra

redundancy or protection; the alternative is costly and time-wasting rework.

- Because software errors and flawed human decision making do not involve random failures, traditional hazard analysis techniques that only identify such failures will not be effective for them.

Against this background, Leveson developed a new accident model called STAMP (Systems-Accident Model and Processes), which has been described in detail elsewhere [2]. The new hazard analysis technique based on STAMP is called STPA (Systems-Theoretic Process Analysis) [3]. STPA views the system as a collection of interacting loops of control. An important advantage of this technique is that it can be used to drive the earliest design decisions. In safety-guided design, the information needed by the designers to make good decisions is provided to them before they create the design and the analyses are performed in parallel with the design process rather than after it. The safety-guided design using STPA is discussed in details in another paper presented at IAASS 2011 [4].

STPA uses functional control structure diagrams and system hazards to generate the system and component safety constraints and safety requirements. When STPA is applied to an existing design, this information is available at the beginning of the analysis process. Figure 1 shows a generalized safety control structure diagram, which does not represent any one particular system.

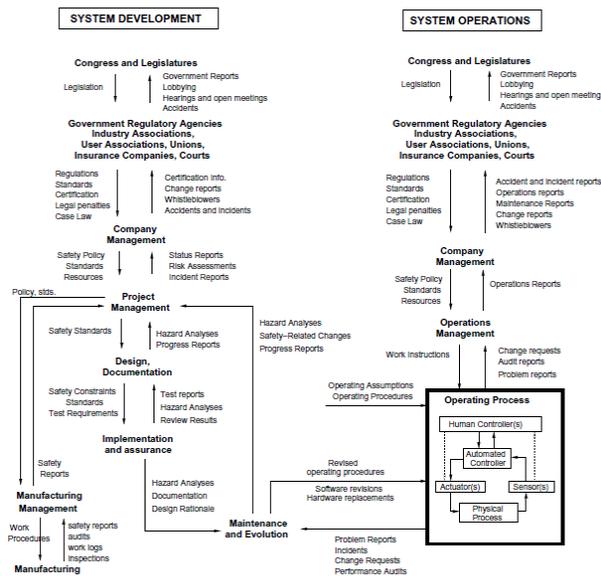


Figure 1. Generalized control structure diagram [5].

STPA has two main steps:

1. Identify the potential for inadequate control of the system that could lead to a hazardous state. Hazardous states result from inadequate control or enforcement of the safety constraints, which can occur because:
  - 1) A control action required for safety is not provided;
  - 2) An unsafe control action is provided that leads to a hazard;
  - 3) A potentially safe control action is provided too early, too late, or out of sequence;
  - 4) A safe control action is stopped too soon (for a continuous or nondiscrete control action) or applied too long.

For convenience, a table that shows the above four unsafe control actions for each control action can be used to record the results of this step.
2. Determine how each potentially hazardous control action identified in step 1 could occur.
  - a) For each unsafe control action, examine the parts of the control loop to see if they could cause it (this can be done using the general causal factors shown in Figure 2 as a guide). Design controls and mitigation measures if they do not already exist or evaluate existing measures if the analysis is being performed on an existing design.
  - b) Consider how the designed controls could degrade over time and build in protection, including
    - i) Management of change procedures to ensure safety constraints are enforced in planned changes.
    - ii) Performance audits where the assumptions underlying the hazard analysis are the preconditions for the operational audits and controls so that unplanned changes that violate the safety constraints can be detected.
    - iii) Accident and incident analysis to trace anomalies to the hazards and to the system design.

While the analysis can be performed in one step, dividing the process into discrete steps provides the safety engineers with a structured process for hazard analysis. The information from the first step (identifying the unsafe control actions) is required to perform the second step (identifying the causes of the unsafe control actions).

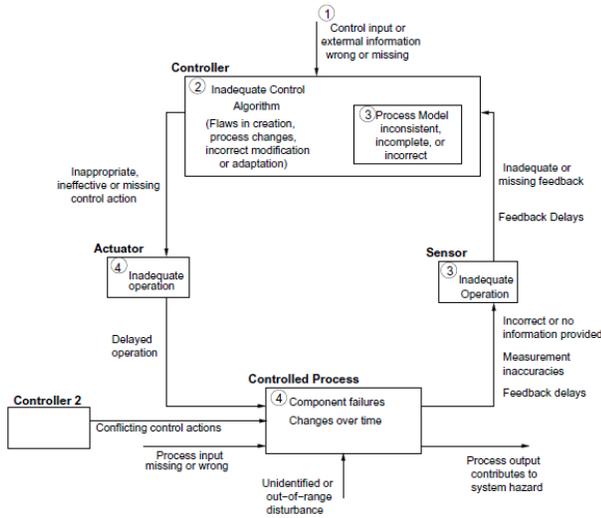


Figure 2. Causal factors to be considered to create scenarios in Step 2 [5].

## 2.2. Extended STPA Step 1

In the first step of STPA described in the previous section, inadequate control actions that could lead to a hazard are divided into four categories: 1) not provided, 2) unsafe provided, 3) provided too early, too late, or out of sequence (wrong timing), and 4) stopped too soon or applied too long (wrong duration). In the multiple-controller situations, however, even if a potentially safe control action is provided by one of the controllers, conflicts or coordination problems with other controllers could occur. As seen in the example of conflicting instructions between TCAS and ATC, each controller thought it provided a “safe” control action but as a result, the lack of coordination between the two controllers provided an unsafe control action to the entire system. Another example is the interference between “safe” control actions. If each controller provides the same “safe” command to the system and the system allows one command to be overridden by another command, then the process might be started over again while time runs out. Therefore, “provided” should also be included in the analysis as another category of inadequate control action as follows:

- 0) A safe control action is provided;
- 1) A control action required for safety is not provided;
- 2) An unsafe control action is provided;
- 3) A potentially safe control action is provided too early, too late, or out of sequence (wrong timing);
- 4) A control action is stopped too soon or applied too long (wrong duration).

Then, as shown in Table 1, interactions of control actions between two controllers can be analyzed in the form of table. Note that for interactions between  $N$  controllers, this tabular form can be extended to  $N$ -dimension. As each cell in the table is labeled with a colored tag, each combination of control actions can then be divided into the following four categories:

- Only one safe control action is provided.
- Multiple safe control actions are provided.
- Both safe and unsafe control actions are provided.
- Only unsafe control actions are provided.

This classification is helpful to understand the level of hazard for each combination. For the blue category, the system should be safe so there is no need to analyze in any more depth. For the green category, because this could lead to a hazard, the system should be designed or controlled such that only one safe control action is properly executed even if redundantly provided. For the yellow category, because this is also potential hazardous, the system should be designed or controlled such that the safe one is properly executed without being interrupted by the unsafe ones. For the red category, because this is obviously hazardous, we need to identify the causal factors in Step 2.

Table 1. Unsafe interactions of control actions between two controllers.

		A Control Action by Controller 1				
		Provided	Not Provided	Unsafe Provided	Wrong Timing	Wrong Duration
A Control Action by Controller 2	Provided	<span style="color: green;">■</span>	<span style="color: blue;">■</span>	<span style="color: yellow;">■</span>	<span style="color: yellow;">■</span>	<span style="color: yellow;">■</span>
	Not Provided	<span style="color: blue;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>
	Unsafe Provided	<span style="color: yellow;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>
	Wrong Timing	<span style="color: yellow;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>
	Wrong Duration	<span style="color: yellow;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>	<span style="color: red;">■</span>

## 2.3. Extended STPA Step 2

Having obtained the list of unsafe interactions of control actions between multiple controllers, the next step is to

identify causal factors arising from multiple controller contributions for each combination of unsafe control actions. In order to determine how each combination could occur, it is important to look at the entire system including other controllers and the controlled process from each controller's point of view. Although such causal factors may vary widely depending on the control structure, we could come up with some general guidewords by thinking about how multiple controllers' control actions could be inconsistent or uncoordinated with each other. The key is a process model<sup>1</sup>. The general guidewords that we have come up with are listed below:

- Misconception of situation
- Miscommunication between controllers
- Overconfidence in automation
- Unclear responsibility
- Unclear authority
- Withholding under other controllers' control actions
- Rash control under pressure/desire to precede other controllers
- Satisfaction by other controllers' control actions
- Confusion by other controllers' unexpected control actions

After several causal factors have been identified for each controller with the assistance of the above general guidewords, a specific hazardous scenario could be built up by selecting any combination of those causal factors. A number of hazardous scenarios could be derived in this way. Once a specific scenario of interest has been found, then each causal factor leading to this scenario should be eliminated or controlled by the design or operations.

While this step of the analysis is more or less ad-hoc, using a control structure diagram with each controller augmented with a process model, along with these guidewords, provides a helpful guideline to identify causal factors arising from multiple controller situations.

### 3. CASE STUDY OF HTV APPROACH PHASE

This section presents the case study of an STPA analysis for multiple controllers on the HTV final approach phase. First, we describe the role of multiple controllers in this phase and organize the information needed for the analysis.

#### 3.1. Overview of HTV Proximity Operations

As shown in Figure 3, 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 (SSRMS) and berthed to the ISS. This operation phase is called PROX Operations. The HTV approach sequence during PROX Operations is as follows:

1. The HTV moves from the AI point to a point 500 [m] below the ISS guided by GPS (RGPS Navigation).
2. While keeping its attitude relative to the ISS using its attitude control system, the HTV approaches the ISS using a laser sensor called Rendezvous Sensor (RVS) to beam the laser to the reflector located on the nadir side of the Kibo module (RVS Navigation).
3. The HTV holds its approach twice: when reaching 250 [m] below the ISS (hold point) and 30 [m] below the ISS (parking point). At the hold point, the HTV performs a 180-degree yaw around to prepare for a Collision Avoidance Maneuver (CAM) in the case that the HTV's relative position is too close or relative approach rate is faster than the predefined threshold.
4. Finally, once the HTV reaches 10 [m] below the ISS, called Capture Box, the HTV maintains its relative position and attitude within the Capture Box and waits for the ISS crew commanding the deactivation of attitude control (*Free Drift*).

While in our past study presented at IAASS 2010 we focused on the capture phase at the Capture Box to demonstrate the application of STPA, in this study of multiple controller problems, we focus on the final approach phase, especially from 30 [m] below the ISS up to the Capture Box. In this phase, the HTV approaches the ISS automatically without any commands by the ISS crew or the ground station (GS) crew in the nominal case. If everything goes well, no controls are provided in this phase. In case of an emergency, however, either the ISS crew or the GS crew has to issue *Hold*, *Retreat*, or *Abort* commands. The details about the control structure and off-nominal commands are described in the next section.

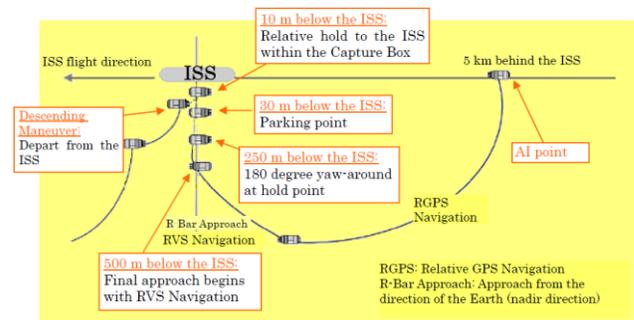


Figure 3. HTV Proximity Operations [6].

<sup>1</sup> Often called a “mental model” for human controllers

### 3.2. Control Structure and Off-Nominal Commands

As shown in Figure 4, the high-level control structure for the HTV final approach is composed of three major components: the ISS crew, the GS crew, and the HTV. For simplicity, the GS here represents both NASA and JAXA ground stations, and Tracking and Data Relay Satellite (TDRS) as a backup communication system is omitted. Connecting lines between the components represent commands or information. There is also a voice loop connection between the ISS crew and the GS crew so that they can communicate with each other through the entire operation. As described in the previous section, the HTV performs an automatic approach during this phase in the nominal case. If an emergency occurs, the ISS crew can send commands such as *Hold*, *Retreat*, and *Abort* to the HTV using the Hardware Command Panel (HCP). These three commands are also available to the GS crew. 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.

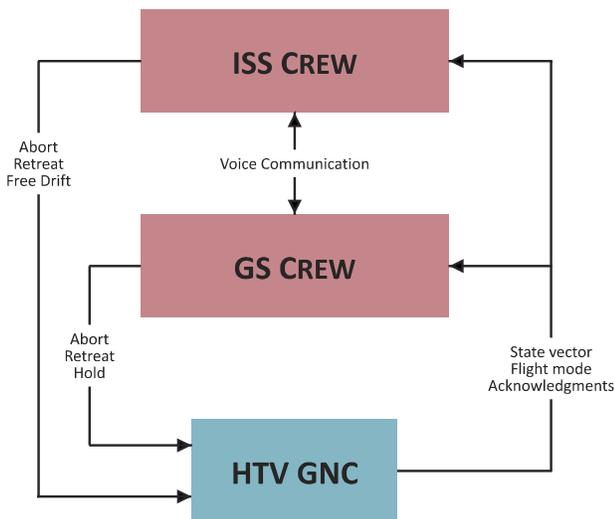


Figure 4. High-level control structure for HTV final approach phase.

Table 2 summarizes the availability of the off-nominal commands to each of the three controllers. These off-nominal commands as well as *Free Drift* are put in Figure 4 as control actions provided by the ISS and GS crew. Table 3 lists the off-nominal command sequence and range, showing that in case of an emergency, the ISS and GS crew is supposed to issue *Hold*, *Retreat*, and *Abort* in the ranges of 30 [m] to 15 [m], 15 [m] to 10 [m], and Capture Box and beyond, respectively, while the HTV 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 defense before the HTV collides with the ISS. For this reason, hereinafter, we are focused on an *Abort* command.

Table 2. Off-nominal command availability.

	<i>Abort</i>	<i>Retreat</i>	<i>Hold</i>
ISS Crew	✓	✓	☒
GS Crew	✓	✓	✓
HTV GNC	✓	✗	✗

✓: allowed to issue (by the design/FR)

☒: not allowed but available

✗: not available (by the software design)

Table 3. Off-nominal command sequence and range.

	Command	Controller	Range
HTV is closer to ISS ↓	<i>Hold</i>	GS crew	30m – 15m
	<i>Retreat</i>	GS crew	15m – 10m
	<i>Retreat</i>	ISS crew	15m – 10m
	<i>Abort</i>	GS crew	10m (CB) –
	<i>Abort</i>	ISS crew	10m (CB) –
	<i>Abort</i>	HTV GNC	Anywhere

CB: Capture Box

### 3.3. Identifying Potentially Hazardous Control Actions (Step 1)

As described in Section 2.2, the first step of the analysis is to identify unsafe interactions of control actions between controllers using a table like Table 1. Because there are three controllers in this case, we need to create a three-dimensional table as shown in Figure 5. Because *Abort* is the most critical command, we consider a situation where the HTV must be aborted immediately and investigate unsafe interactions or conflicts of *Abort* attempts by the three controllers. The ISS crew could incorrectly provide *Retreat* or *Free Drift* instead of *Abort* while the GS crew could incorrectly provide *Retreat* or *Hold* instead of *Abort*. The HTV GNC is allowed *Abort* only but still could be unsafe by executing an inappropriate maneuver. So here we focus on the three unsafe control actions: 0) *Abort* is

provided, 1) *Abort* is not provided, and 2) an unsafe control action is provided.

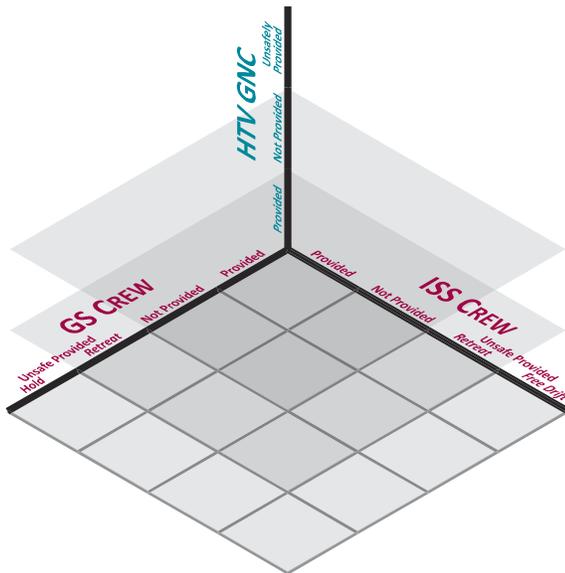


Figure 5. Three-dimensional table to identify unsafe interactions of control actions between the three controllers.

Tables 4-6 at the end of this paper show the various hazardous interactions identified. Each cell in the table describes what could happen as a result of each combination of control actions, assigned an identifier number from (0.0.0) to (2F.2H.2) and labeled with a colored tag of the four categories. The summary is:

- Only one safe control action is provided:  
(1.1.0), (1.0.1), (0.1.1)
- Multiple safe control actions are provided:  
(0.0.0), (1.0.0), (0.1.0)\*, (0.0.1)
- Both safe and unsafe control actions are provided:  
(2R.0.0), (2F.0.0), (2R.1.0)\*, (2F.1.0), (0.2R.0), (1.2R.0),  
(2R.2R.0), (2F.2R.0), (0.2H.0), (1.2H.0), (2R.2H.0),  
(2F.2H.0), (2R.0.1), (2F.0.1), (0.2R.1), (0.2H.1), (0.0.2),  
(1.0.2), (2R.0.2), (2F.0.2), (0.1.2), (0.2R.2), (0.2H.2)
- Only unsafe control actions are provided:  
(1.1.1)\*, (2R.1.1)\*, (2F.1.1), (1.2R.1), (2R.2R.1),  
(2F.2R.1), (1.2H.1), (2R.2H.1), (2F.2H.1),  
(1.1.2), (2R.1.2), (2F.1.2), (1.2R.2), (2R.2R.2), (2F.2R.2),  
(1.2H.2), (2R.2H.2), (2F.2H.2)

These identifier numbers are assigned in the form of XYZ coordinate in the three-dimensional table in Figure 5. The

X-coordinate represents the ISS crew, the Y-coordinate the GS crew, and the Z-coordinate the HTV GNC. The coordinate value represents one of the three unsafe control actions. For example, the hazard (2R.1.0) corresponds to the combination of 2R) an unsafe control action (*Retreat*) is provided by the ISS crew, 1) *Abort* is not provided by the GS crew, and 0) *Abort* is provided by the HTV GNC”.

Because the blue category should be safe (whether it is coordinated or uncoordinated), we do not consider it anymore. While the red group definitely needs to be investigated further, the green and yellow groups still could lead to a hazard as well. Therefore, as representative examples to present further analyses in Step 2, we pick up four cases, which are starred in the above summary.

### 3.4. Determining How Unsafe Control Actions Could Occur (Step 2)

Figure 6 at the end of this paper shows a detailed control structure, each controller augmented with a process model. Representative hazards selected for this step of the analysis are: (0.1.0), (2R.1.0), (2R.1.1), and (1.1.1). On each of these cases, we discuss how each combination of control actions could occur, paying attention to causal factors arising from multiple controller contributions. Causal factors that are not related to multiple controller contributions are not included in the following analysis because they can be identified by the original STPA. Our focus is on causes that would have not been there in a single-controller situation.

The Step 2 analyses of the four cases are shown below in order of increasing level of hazard. Each bulleted item represents a causal factor of the controller’s unsafe control action that is contributed by other controllers. The general guideword used to identify each causal factor is underlined in brackets. Note that any combinations from bulleted items of each controller could lead to the hazard.

■ (0.1.0) Double *Aborts* are redundantly provided by the ISS crew and the HTV GNC. The command conflict might reset or cancel the Abort procedure.

**ISS Crew**

- Seeing the HTV's "unexpected" self-aborting, the ISS crew thinks that it is floating out of the Abort corridor and issues *Retreat* hastily.  
(Confusion by other controllers' unexpected control actions)

**GS Crew**

The GS crew knows through the voice loop connection that the ISS crew has issued *Abort* and does not issue another *Abort*.

**HTV GNC**

The HTV detects that it needs to abort and initiates a self-abort.

■ (2R.1.0) *Abort* and *Retreat* are provided. If the *Retreat* could override the *Abort*, the HTV might collide with the ISS because the *Retreat* action might not be safe enough.

**ISS Crew**

- The ISS crew thinks that *Retreat* is still safe enough due to incorrect/delayed state vector feedback and issues *Retreat*. (Crew process model inconsistency)
- The ISS crew does not want to waste time and fuel by starting all over again and issues *Retreat* when *Abort* is appropriate before the HTV initiates a self-abort. (Rash control under pressure/desire to precede other controllers)
- Seeing the HTV's "unexpected" self-aborting, the ISS crew thinks that it is floating out of the Abort corridor and issues *Retreat* hastily.  
(Confusion by other controllers' unexpected control actions)

**GS Crew**

- Since *Retreat* has been provided by the ISS crew, the GS crew holds back from any control action and waits and sees for a while.  
(Withholding under other controllers' control actions)
- The GS crew is satisfied with *Retreat* provided by the ISS crew and no longer pays close attention.  
(Satisfaction by other controllers' control actions)
- The GS crew is confused by the ISS crew's unexpected control action and does not issue *Abort*.  
(Confusion by other controllers' unexpected control actions)

**HTV GNC**

The HTV detects that it needs to abort and initiates a self-abort.

■ (2R.1.1) Only *Retreat* is provided. The HTV might collide with the ISS because the *Retreat* action might not be safe enough.

#### ISS Crew

- The ISS crew thinks that *Retreat* is still safe enough due to incorrect/delayed state vector feedback and issues *Retreat*. (Crew process model inconsistency)
- The ISS crew does not want to waste time and fuel by starting all over again and issues *Retreat* when *Abort* is appropriate before the HTV initiates a self-abort. (Rash control under pressure/desire to precede other controllers)

#### GS Crew

- Since *Retreat* has been provided by the ISS crew, the GS crew holds back from any control action and waits and sees for a while. (Withholding under other controllers' control actions)
- The GS crew is satisfied with *Retreat* provided by the ISS crew and no longer pays close attention. (Satisfaction by other controllers' control actions)
- The GS crew is confused by the ISS crew's unexpected control action and does not issue *Abort*. (Confusion by other controllers' unexpected control actions)

#### HTV GNC

- The HTV makes an error in decision-making on self-abort. (HTV component failure)
- The HTV thinks that it is still in a safe position due to inaccurate measurement and that it does not need to self-abort. (Measurement inaccurate)
- The HTV thinks that it is still in a safe position due to delayed state vector feedback and that it does not need to self-abort. (Feedback delayed)
- The HTV thinks that it is still in a safe position due to incorrect state vector feedback and that it does not need to self-abort. (Feedback incorrect)
- Since *Retreat* has been provided by the ISS crew, the HTV GNC refrains from self-aborting. (Withholding under other controllers' control actions)

■ (1.1.1) No *Abort* is provided. The HTV is left in an unsafe situation and might collide with the ISS.

#### ISS Crew

- Seeing the HTV float out of the *Abort* corridor, the ISS crew thinks that the HTV is aborting and that they do not need to issue *Abort*. (Misconception of situation)
- Miscommunicating with the GS crew through the voice loop connection, the ISS crew thinks the GS crew has issued *Abort* and vice versa. (Miscommunication between controllers)
- Overestimating the reliability of HTV's self-abort, the ISS crew relaxes too much and does not pay close attention. (Overconfidence in automation)
- The ISS crew thinks the GS crew will issue *Abort* and vice versa. (Unclear responsibility)
- Waiting for the directive from or confirmation with the GS crew, the ISS crew hesitates to issue *Abort*. (Unclear authority)

#### GS Crew

- Judging from the telemetry from the HTV floating out of the *Abort* corridor, the GS crew thinks that the HTV is self-aborting and that they do not need to issue *Abort*. (Misconception of situation)
- Miscommunicating with the ISS crew through the voice loop connection, the GS crew thinks the ISS crew has issued *Abort* and vice versa. (Miscommunication between controllers)
- Overestimating the reliability of HTV's self-abort, the GS crew relaxes too much and does not pay close attention. (Overconfidence in automation)
- The GS crew thinks the ISS crew will issue *Abort* and vice versa. (Unclear responsibility)
- Waiting for the directive from or confirmation with the NASA GS crew, the JAXA GS crew hesitates to issue *Abort* and vice versa. (Unclear Authority)

#### HTV GNC

- The HTV makes an error in decision-making on self-abort. (HTV component failure)
- The HTV thinks that it is still in a safe position due to inaccurate measurement and that it does not need to self-abort. (Measurement inaccurate)
- The HTV thinks that it is still in a safe position due to delayed state vector feedback and that it does not need to self-abort. (Feedback delayed)
- The HTV thinks that it is still in a safe position due to incorrect state vector feedback and that it does not need to self-abort. (Feedback incorrect)

As shown above, several causal factors have been identified for each controller using the general guidewords listed in Section 2.3. Any combination of these causal factors could be a hazardous scenario. For example, one possible scenario leading to the hazard (2R.1.1) is:

- The ISS crew does not want to waste time and fuel by starting all over again and issues *Retreat* when *Abort* is appropriate before the HTV initiates a self-abort. (Rash control under pressure/desire to precede other controllers)
- The GS crew is satisfied with *Retreat* provided by the ISS crew and no longer pays close attention. (Satisfaction by other controllers' control actions)
- Since *Retreat* has been provided by the ISS crew, the HTV GNC refrains from self-aborting. (Withholding under other controllers' control actions)

This way, a number of such hazardous scenarios could be derived from combinations of causal factors. In order to avoid those scenarios, each causal factor should be eliminated or controlled by the design or the flight rules though mental factors of human controllers such as “rash” control or “satisfaction” are difficult to control in the design or the flight rules.

### 3.5. Discussions

Some of the causal factors that are identified are those that were lumped together as “process model inconsistency” in the original STPA. In other words, this step refines and details the process model inconsistency arising from multiple controller contributions. However, this case study, where there are two human controllers and one automated controller, is just one case among many other possible configurations. Therefore, the outcome here might be weighted toward causal factors arising between human controllers. Problems that are likely to be unique to multiple automated controller situations (e.g., getting into an infinite loop) do not appear here. Additionally, we only looked at discrete control actions such as *Abort* and *Retreat* commands. For continuous or nondiscrete control actions, while one controller is performing something, another controller could cut in (intervene). For these reasons, there would still be other general guidewords leading to multiple-controller-related hazards that could not be covered in this case study.

## 4. SUMMARY AND FUTURE WORK

Following the demonstration of a new hazard analysis technique, STPA, on the HTV capture phase, which was presented at IAASS 2010, we explored multiple controller contributions to hazards using the HTV final approach

phase. We defined a procedure to identify possible unsafe interactions among controllers. The procedure includes:

- Adding the 0<sup>th</sup> category of inadequate control actions: “A correct or safe control action is provided”
- Examining unsafe interactions between  $N$  controllers' control actions using an  $N$ -dimensional table
- Classifying the identified interactions into four categories of level of hazard
- Identifying multiple-controller-related causal factors using the general guidewords and a control structure diagram augmented with process models
- Building up a scenario leading to the hazard from any combinations of the identified causal factors for each controller

Some of the results refined and detailed the “process model inconsistency” arising from multiple controller contributions.

This case study does not cover many other possible configurations of a mix of human and automated controllers. For this reason, there are other general guidewords leading to multiple-controller-related hazards that were not covered in this case study. Therefore, future work is needed to further refine this method by applying it to other cases. As an additional way of identifying causal factors, we will also study how to analyze interferences between controllers using mathematical engines such as SpecTRM. On the HTV side, while the outcome of this study might not be something explicitly written in the hazard report, the results should be provided as a recommendation to the current operational procedures of the HTV missions.

## 5. ACKNOWLEDGMENTS

The authors would like to thank members of JAXA, JAMSS, and the Complex Systems Research Laboratory (CSRL) at MIT for their support in this study. This research was performed at MIT and JAXA under a contract with JAMSS.

## 6. ACRONYMS

CAM: Collision Avoidance Maneuver  
CB: Capture Box  
CMD: Command  
GNC: Guidance Navigation and Control  
GS: Ground Station  
HCP: Hardware Command Panel  
HTV: H-IIB Transfer Vehicle  
ISS: International Space Station  
OBS: On-Board Software

OCS: Operations Control System  
RCS: Reaction Control System  
RVS: Rendezvous Sensor  
SpecTRM: Specification Tools and Requirements Methodology  
SSRMS: Space Station Remote Manipulator System  
STAMP: Systems-Theoretic Accident Model and Processes  
STPA: Systems-Theoretic Process and Analysis  
TDRS: Tracking and Data Relay Satellite

- Based Specifications.” *Proceedings of the 5<sup>th</sup> IAASS Conference*, Versailles, France, October 17-19, 2011.
- Leveson, N., *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press, 2011.
  - Japan Aerospace Exploration Agency, “HTV 2 (KOUNOTORI 2) Mission Press Kit,” January 2011.

**7. REFERENCES**

- Ishimatsu, T., Leveson, N., Thomas, J., Katahira, M., Miyamoto, Y., and Nakao, H., “Modeling and Hazard Analysis using STPA,” *Proceedings of the 4<sup>th</sup> IAASS Conference*, Huntsville, AL, May 19-21, 2010.
- Leveson, N., “A New Accident Model for Engineering Safer Systems,” *Safety Science*, Vol. 42, No. 4, pp. 237-270, April 2004.
- Leveson, N., “Software Challenges in Achieving Space Safety,” *Journal of the British Interplanetary Society (JBIS)*, Volume 62, 2009.
- Fleming, C., Ishimatsu, T., Miyamoto, Y., Nakao, H., Katahira, M., Hoshino, N., Thomas, J., and Leveson N., “Safety Guided Spacecraft Design using Model-

Table 4. Unsafe interactions of control actions between the three controllers (Abort provided by HTV).

ISS GS		Abort Provided by HTV GNC			
		Abort by ISS CREW		Unsafe Provided	
		Provided	Not Provided	Retreat	Free Drift
Abort by GS CREW	Provided	(0.0.0) Triple Aborts are redundantly provided by ISS, GS, and HTV. The command conflict might reset or cancel the Abort procedure.	(1.0.0) Double Aborts are redundantly provided by GS and HTV. The command conflict might reset or cancel the Abort procedure.	(2R.0.0) Duplicated Aborts and Retreat are provided. If Retreat could override Abort, the HTV might collide with the ISS because the Retreat action might not be safe enough.	(2F.0.0) Duplicated Aborts and Free Drift are provided. If Free Drift could override Abort, the HTV might collide with the ISS because the HTV becomes a free-flying object.
	Not Provided	(0.1.0) Double Aborts are redundantly provided by ISS and HTV. The command conflict might reset or cancel the Abort procedure.	(1.1.0) Whether it's coordinated or uncoordinated, the HTV is provided a single Abort and will be safely aborted.	(2R.1.0) Abort and Retreat are provided. If Retreat could override Abort, the HTV might collide with the ISS because the Retreat action might not be safe enough.	(2F.1.0) Abort and Free Drift are provided. If Free Drift could override Abort, the HTV might collide with the ISS because the HTV becomes a free-flying object.
	Retreat	(0.2R.0) = (2R.0.0).	(1.2R.0) = (2R.1.0).	(2R.2R.0) Abort and two incorrect control actions are provided. If either one could override Abort, the HTV might collide with the ISS because the executed action might not be safe enough.	(2F.2R.0) = (2R.2R.0).
	Hold	(0.2H.0) Duplicated Aborts and Hold are provided. If Hold could override Abort, the HTV might collide with the ISS because the Hold action might not be safe enough.	(1.2H.0) Abort and Hold are provided. If Hold could override Abort, the HTV might collide with the ISS because the Hold action might not be safe enough.	(2R.2H.0) = (2R.2R.0).	(2F.2H.0) = (2R.2R.0).

Table 5. Unsafe interactions of control actions between the three controllers (Abort not provided by HTV).

ISS		Abort Not Provided by HTV GNC			
		Abort by ISS CREW			
		GS	Provided	Not Provided	Unsafe Provided
Retreat	Free Drift				
Abort by GS CREW	Provided	(0.0.1) Double Aborts are redundantly provided by ISS and GS. The command conflict might reset or cancel the Abort procedure. 	(1.0.1) Whether it's coordinated or uncoordinated, the HTV is provided a single Abort and will be safely aborted. 	(2R.0.1) Abort and Retreat are provided. If Retreat could override Abort, the HTV might collide with the ISS because the Retreat action might not be safe enough. 	(2F.0.1) Abort and Free Drift are provided. If Free Drift could override Abort, the HTV might collide with the ISS because the HTV becomes a free-flying object. 
	Not Provided	(0.1.1) = (1.0.1). 	(1.1.1) No Abort is provided. The HTV is left in an unsafe situation and might collide with the ISS. 	(2R.1.1) Only Retreat is provided. The HTV might collide with the ISS because the Retreat action might not be safe enough. 	(2F.1.1) Only Free Drift is provided. The HTV might collide with the ISS because the HTV becomes a free-flying object. 
	Unsafe Provided Retreat	(0.2R.1) = (2R.0.1). 	(1.2R.1) = (2R.1.1). 	(2R.2R.1) Two incorrect control actions are provided. Even if either one is executed, the HTV might collide with the ISS because the executed action might not be safe enough. 	(2F.2R.1) = (2R.2R.1). 
	Unsafe Provided Hold	(0.2H.1) Abort and Hold are provided. If Hold could override Abort, the HTV might collide with the ISS because the Hold action might not be safe enough. 	(1.2H.1) Only Hold is provided. The HTV might collide with the ISS because the Hold action might not be safe enough. 	(2R.2H.1) = (2R.2R.1). 	(2F.2H.1) = (2R.2R.1). 

Table 6. Unsafe interactions of control actions between the three controllers (Abort incorrectly provided by HTV).

ISS		Abort Unsafely Provided by HTV GNC			
		Abort by ISS CREW			
		GS	Provided	Not Provided	Unsafe Provided
Retreat	Free Drift				
Abort by GS CREW	Provided	(0.0.2) Double Aborts and incorrect Abort are provided. If incorrect Abort could override the other Aborts, the HTV might collide with the ISS because of the incorrect Abort action. 	(1.0.2) Abort and incorrect Abort are provided. If incorrect Abort could override the other Abort, the HTV might collide with the ISS because of the incorrect Abort action. 	(2R.0.2) Abort and two incorrect control actions are provided. If either one could override Abort, the HTV might collide with the ISS because the executed action might not be safe enough. 	(2F.0.2) = (2R.0.2). 
	Not Provided	(0.1.2) = (1.0.2). 	(1.1.2) Only incorrect Abort is provided. The HTV might collide with the ISS because of the incorrect Abort action. 	(2R.1.2) Two incorrect control actions are provided. Even if either one is executed, the HTV might collide with the ISS because the executed action might not be safe enough. 	(2F.1.2) = (2R.1.2). 
	Unsafe Provided Retreat	(0.2R.2) = (2R.0.2). 	(1.2R.2) = (2R.1.2). 	(2R.2R.2) Three incorrect control actions are provided. Even if either one is executed, the HTV might collide with the ISS because the executed action might not be safe enough. 	(2F.2R.2) = (2R.2R.2). 
	Unsafe Provided Hold	(0.2H.2) = (2R.0.2). 	(1.2H.2) = (2R.1.2). 	(2R.2H.2) = (2R.2R.2). 	(2F.2H.2) = (2R.2R.2). 

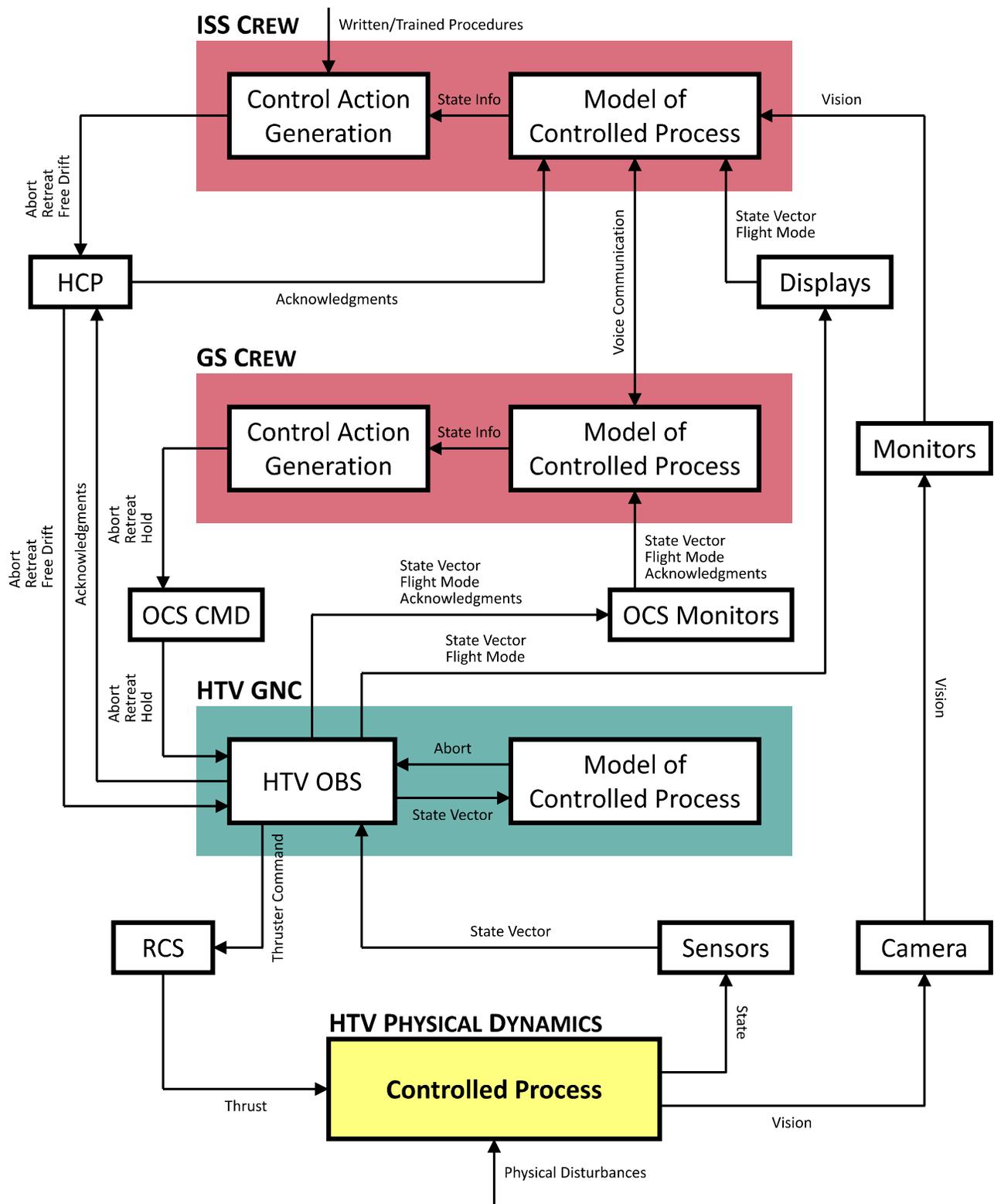


Figure 6. Control structure augmented with process models for HTV final approach phase.