JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 1, December 2004

An Autonomous Software Safety System for a Dexterous Space Robot

Stephen Roderick,[*] Brian Roberts,[†] Ella Atkins,[‡] Phil Churchill,[§] and David Akin[**]
University of Maryland, College Park, Maryland 20742

This paper describes a fail-safe software-based hazard control system designed for safety-critical operation of a dexterous space robot. This software safety system is fully autonomous, requiring no human intervention to detect or prevent a hazardous situation. Using a system analysis, we show that a computer-based approach is necessary due to issues of time delay, inadequate human reaction time given robot tip velocities, and the complexity of the robot and workspace. A system-wide software design is presented that satisfies the constraints of this analysis. The validation of this safety-critical software system is examined with fault trees constructed to determine combinations of events necessary to cause a hazard. These fault trees are qualitatively and quantitatively evaluated to determine the minimal set of events that can cause a hazard and the probability of a hazard occurring. A sensitivity analysis is conducted to determine which failures, and which combination of failures, most contribute to hazard probabilities. The theoretical analysis and sensitivity results have been presented previously, however this paper extends this work by examining the operating history of the robot, presenting its safety record, and discussing the system failures that have occurred.

I. Introduction

Robotic servicing and structural assembly systems have been proposed to extend the life of Earth-orbiting spacecraft such as the Hubble Space Telescope (HST) and to support future exploration of the Moon and Mars. The University of Maryland Space Systems Laboratory (SSL) has developed dexterous robotic systems capable of performing in-space servicing and assembly operations, and has tested these systems with complex end-to-end servicing tasks in a neutral buoyancy space simulation environment. Based largely on knowledge developed from analysis and neutral buoyancy simulation of HST servicing, in 1992 the Ranger satellite servicing system was designed with a goal of using the same extravehicular activity (EVA) interfaces that the astronauts use when they work in space, enabling the robot to accomplish the same tasks currently performed by EVA astronauts without the need for specialized robotic interfaces at the worksite.

The shuttle-based version of Ranger - the Ranger Telerobotic Shuttle Experiment (TSX) - shown in Fig. 1, consists of two 8 degree-of-freedom (DOF) dexterous manipulators with two tool drives, a 6-DOF positioning leg that attaches the robot head to a space shuttle payload bay fixture, and a 7-DOF video manipulator that provides positioning and direction of a stereo camera pair for operator feedback. The two dexterous arms, shown in Fig. 2, have greater range, speed, and force of motion than even an unsuited human. A neutral buoyancy version of Ranger TSX, shown in Fig. 3, was built for training of the flight operators and is operational in both 1-g and neutral buoyancy environments.

This paper is part of the December Special Section on Intelligent Systems. Received 18 August 2004; revision received 1 November 2004; accepted for publication 21 November 2004.

[*] Faculty Research Assistant, Department of Aerospace Engineering. AIAA Member.

[†] Faculty Research Assistant, Department of Aerospace Engineering. AIAA Member.

[‡] Assistant Professor, Department of Aerospace Engineering. Senior Member, AIAA.

[§] Senior Software Engineer, ATI Research, Inc. Formerly of the Univ. of Maryland.

[**] Associate Professor, Department of Aerospace Engineering. Senior Member, AIAA.

To avoid the complexity of an anthropomorphic hand, the Ranger approach was to incorporate autonomously interchangeable end effectors that are actuated by two tool drives at the end of the dexterous arms. This provides an almost unlimited range of potential interfaces, and high mission reliability through the development of a mechanical interchange mechanism that ensures the end effector is never released in the exchange between the tool holder and the robot arm. The entire system was designed to operate via ground control with two-way command and data relay through NASA's Tracking and Data Relay Satellite system. Software developed for the ground control station.[1] provided proven mitigation of the 3-7 second command latency through the use of predictive and commanded displays.

This paper presents the results of a hazard analysis for the Ranger TSX experiment and describes Ranger's shuttle-certified autonomous hazard control system that was designed to cope with these hazards, enabling safe operation in close proximity to hardware not designed to withstand a high-speed manipulator impact. This hazard control system was a requirement for flying on the shuttle and has successfully completed three of the four phases of shuttle payload certification, however, the analysis discussed in this paper was not required as part of the payload certification process. Through characterization of fault scenarios and their likelihood, software was developed to constrain manipulator motions or halt all motion when the system identifies a potential hazard. Validation of this hazard control software is performed via fault tree analysis, including both qualitative and quantitative evaluations. Due to budget restrictions following the discovery of International Space Station cost overruns and difficulties in finding flight opportunities in the shuttle manifest, the Ranger program was canceled by NASA therefore it will never fly on the shuttle, however, parts of the system have been operational for over two years, enabling analysis of system performance over a spectrum of 1-g and neutral buoyancy tests with experienced and novice operators. The operational history of this experiment is summarized, and expected hazard probabilities are contrasted with the actual operational data. This paper concludes with a summary of ongoing work to better model failure probabilities and to adapt the hazard control system for dexterous robotic systems beyond Ranger.

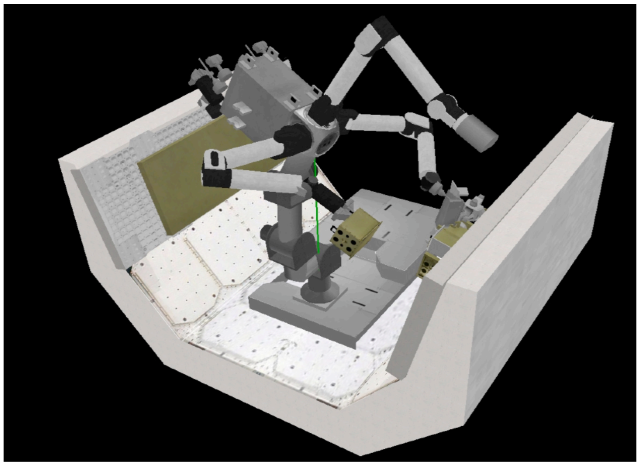

Fig. 1 Ranger telerobotic shuttle experiment.

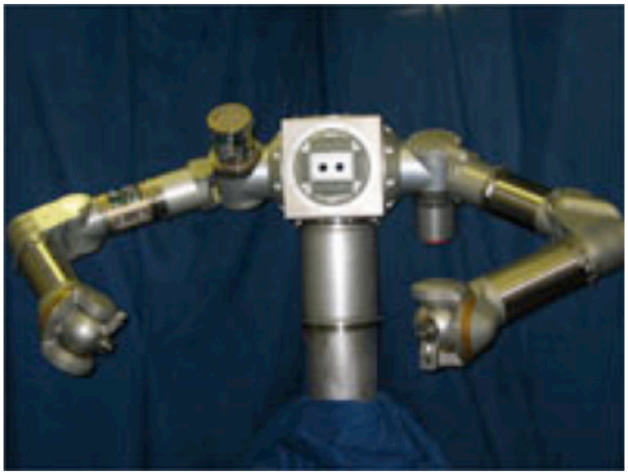

Fig. 2 Ranger dexterous manipulators.

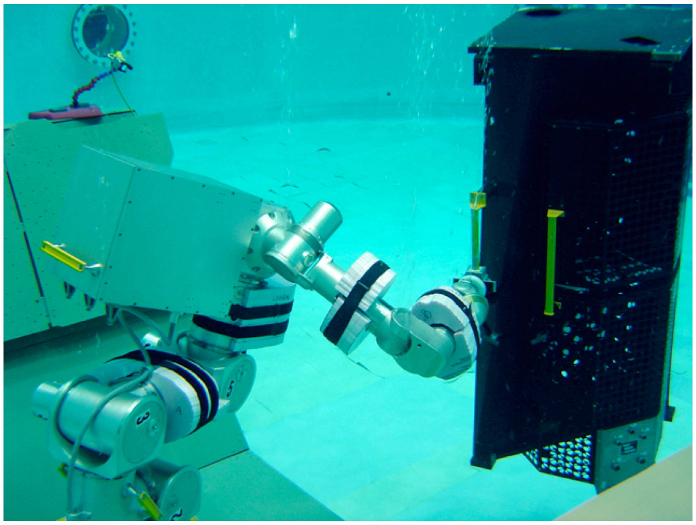

Fig. 3 Ranger satellite servicing system neutral buoyancy robot.

II. Definitions and Background

There are many different definitions of safety and related terms such as *fault* and *failure*. For consistency, the definitions used by this paper are taken from Refs. 2 and 3.

A *failure* is an abnormal occurrence.

A *fault* is a higher-order event caused by one or more failures.

An *accident* is an undesired and unplanned event, resulting in a level of loss. Loss is a general concept and the actual type of loss is system-specific, but can include injury to humans or damage to property.

A *hazard* is a system state and other environmental conditions, which inevitably lead to an accident.

Hazard likelihood is the qualitative or quantitative probability of the hazard occurring.

Safety is freedom from accidents.

Designing a system to be free of all accidents, no matter how perverse or remote, may require so many compromises in functionality that the system is not worth building at all. Safety is not an absolute concept; a system can only be built to reduce the risk of an accident to an acceptable level.[4,5] Safety is also an attribute of the entire system; it is not driven by only certain components of the system. This requires that analyses of system safety include all components: software, hardware, and operators.[2,6,7]

Numerous safety-critical software systems have been developed and deployed in domains ranging from aircraft flight management systems[8] to nuclear power plants.[9] Fault tree analysis methods similar to that adopted for Ranger hazard control are a standard and accepted industry practice when identifying and characterizing the likelihood of hazards.[10] Ranger manipulator systems, as discussed above, require handling features unique to the space environment (0-g, radiation) as well as operation in close proximity to non-contactable objects (i.e., objects designed to withstand only astronaut kick loads) that could be damaged by Ranger's high-speed, high-torque manipulators.

Terrestrial robots have faced similar problems to that required of the hazard control system presented here. The context of the work to be performed in space introduces several new aspects:

Safing a manipulator may not simply mean removing motor power. In a terrestrial setting this may be incorrect as a massive manipulator would then fall under gravity's influence and potentially damage anything underneath the manipulator. In space, removing motor power causes the manipulator to cease any motion (typically there will be a small amount of drift, which is rapidly damped out through internal friction and the back electromagnetic forces generated by the motor. On Ranger TSX, analysis indicates a drift of no more than a few centimeters).

A lack of gravity implies that objects that are let go do not simply drop to the floor as they would in a terrestrial setting. This adds the hazard of releasing an untethered object. Detachable objects cannot simply be let go as they may bounce around in the payload bay, damaging the shuttle.

Few terrestrial robot systems are exposed to as harsh an environment as space. Deleterious effects include substantial launch and landing loads, electronic-damaging radiation, severe fluctuations in thermal conditions, and an inability to service the robot directly.

III. System hazard analysis

A preliminary hazard analysis (PHA)[2,11,12] was conducted on the Ranger TSX experiment that identified the following three hazards that Ranger presents to the shuttle and its crew.

Hazard A. Manipulator motion physically damages the shuttle or prevents a safe return to Earth (e.g. by preventing the payload bay doors from closing). This could be caused by a manipulator end-point, an object grasped by a manipulator, or any other portion of a manipulator or the Ranger body impacting the shuttle or violating the payload bay door envelope.

Hazard B. Releasing an untethered object (e.g. an orbital replacement unit) that damages the shuttle or becomes orbital debris. This would occur when a gripper incorrectly opens while grasping an orbital replacement unit (ORU) that is not constrained by its receptacle.

Hazard C. An object (e.g. an ORU restraint bolt) breaks due to excessive force or torque, and the subsequent pieces damage the shuttle or become orbital debris.

All of these hazards can be considered to be caused by motion of a motor: physically damaging the shuttle requires motion of the manipulators, releasing an object requires the motors to open the grippers, and applying torque requires motor motion.

There were three possible hazard control options for this mission; electro-mechanical, human, and computer-based. Electro-mechanical hazard controls could involve a) caging the robot, substantially increasing mass; b) placing the robot out of reach of non-contactable objects, which by definition implies that the robot can still contact whatever is placing it out of reach; c) reducing the robot's capabilities so that it can not impart excessive energy, which prevents the robot from being able to accomplish its tasks, and d) tethering individual task equipment, which may cause more of a hazard due to interference with other operations. Because each of these options had significant drawbacks, the use of purely mechanical or electrical controls for the Ranger mission was infeasible.

The reaction time needed to prevent a failure from causing part of Ranger to impart excess energy into a non-contactable object is a function of: which manipulators are operating, the inertia seen by the manipulator, the maximum operating velocity of the manipulator, and the distance the manipulator will take to drift to a stop when power is removed from its actuators. The maximum reaction time can be found by forcing the failure to accelerate the robot from a zero initial velocity,[13] as shown in Fig. 4. If the manipulator was already in motion, then less time would be required to reach the maximum operating velocity and hence the system would have even less time to react to the failure. The maximum permissible reaction times shown here are 0.056 seconds for the dexterous manipulators and 0.139 seconds for the positioning leg. The minimum human reaction time under optimal conditions is 0.12 to 0.15 seconds, while for spontaneous movement it may be several hundred milliseconds (Ref. 14). On this basis, even with no teleoperation-related time delay, the use of a human for hazard control for the Ranger mission is infeasible.

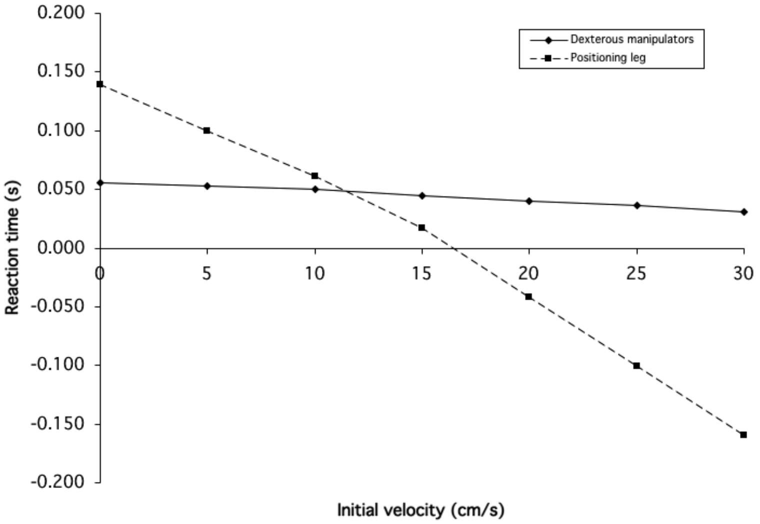

Fig. 4 Required reaction time for an initial operating velocity, such that no more than the maximum energy is imparted to a non-contactable object.

A computer-based hazard control system must have sufficient data available to detect failures, sufficient authority available at the actuators to safe the system, and must react with sufficient speed. As will be shown, the Ranger software safety system satisfies these constraints and so is the only feasible hazard control system. This hazard control system also was designed to satisfy NASA requirements for space shuttle payloads.

NASA has developed a set of criteria[15] that must be met by payloads operating on manned spacecraft such as the shuttle. This traditional approach ensures that a payload can tolerate some minimum number of credible failures and/or operator errors without creating a hazardous event. This traditional approach was infeasible for Ranger TSX, however, because it required at least one hazard control method to be independent of a computer system for a catastrophic hazard (a hazard that could result in fatal personnel injury or loss of the shuttle). With a vision toward more complex computer-based payloads, NASA has defined an alternative “fail-safe” hazard control approach[16] that allows a computer-based control system to have total control of a hazardous payload when the traditional approach is infeasible. A “fail-safe” control system can be interrupted without resulting in a hazard, that is the system can be placed into a “safe” state from which it cannot cause a hazard. Note that this does not necessarily mean that the system can continue to function.

The fundamental precept of this fail-safe approach is that “the control system must *reliably detect the first failure* and *transition the system to a safe state*”. The system need not necessarily be one-fault tolerant, nor does it have to cope with failures subsequent to the first. It simply has to be able to reliably attain a safe state despite the presence of any one failure. This approach allows a computer-based system to be solely and totally responsible for controlling a hazardous payload.

IV. Autonomous Safety System Design

The computer architecture of Ranger’s control system is shown in Fig. 5. There are three local processing units (LPU) in each manipulator, responsible for joint level control of local sensors and actuators. The two power management units (PMU) control the vehicle’s power relays. The two data management units (DMU) communicate with the operator control stations and conduct all vehicle operations. Together they comprise the data management system (DMS). The two DMUs operate control loops at 150 Hz (6.66 ms cycle), the LPUs and PMUs operate at 500 Hz (2 ms cycle), while the communication rate between all computers is 150 Hz (6.66 ms cycle). All computers also operate 7ms watchdog and communication timeouts.

There are two system states: *safe* and *operational*. In the *operational* state manipulator motion is possible. The *safe* state is the state to which the control system transitions when a failure is detected. In this state no commanded motor motion is possible, as the two PMUs have opened all power relays, and each LPU electronically inhibits its associated motors from drawing power. No uncommanded motor motion may also occur, even though the robot may have any physical configuration of its manipulators while in this state. The only external forces that could lead to uncommanded manipulator motion are firings of the shuttle’s control thrusters, however, analysis has shown that friction in the dexterous manipulators and the brakes in the positioning leg are sufficient to maintain the robot configuration in these instances. Also, the grippers used by the manipulator to grasp objects are non-backdriveable and once power is removed from them they will neither open nor close. In this state, no hazard could spontaneously occur.

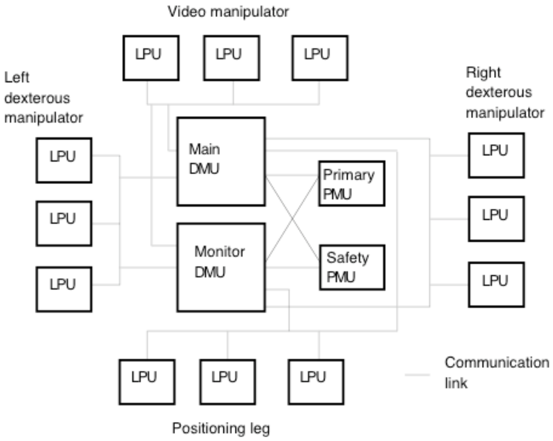

Fig. 5 Architecture of Ranger control system.

A. Failure Identification

A failure of any system component could lead to a hazard occurring. Failure of a sensor can cause a hazard as the control system no longer has an accurate model of the world. Failure of an actuator can cause a hazard as the intended action of the control system is not executed correctly. Failure on the operator's behalf can cause a hazard by executing a hazardous function. The control system, however, cannot cause a hazard alone, it can only cause a hazard through the instruments it controls.[7,12] Failures of each of these components can be categorized as follows:

Hardware. Includes failures of sensors, actuators, and input-output operations.

Software. Includes code defects and upsets due to environmental effects (i.e. radiation in Ranger's case).

Communication. When two or more processors fail to communicate correctly or prevent other computers from being informed of a failure or impending hazard.

Operator. The operator attempts to execute a command that would result in a hazard.

B. Safety Critical Functionality

Computers onboard the robot make all decisions regarding safety; the control stations and the operator do not participate in safety. This simplifies the verification process by removing the control station and communications subsystem from safety considerations. This decision was also made because hardware failure of control station input devices is indistinguishable from a malicious operator attempting to drive the robot to cause a hazard. Relying on operator input via a control station to initiate potentially hazardous functions introduces the possibility of inadvertent execution of said functions. Thus, this system needs only to consider the computers and software onboard the robot.

The onboard safety systems utilize only the actual telemetry of the vehicle in determining whether a hazard is imminent; the computers do not attempt to process, filter and reject operator commands that would cause a hazard. Given the complexity of the workspace and the complexity and speed of the dexterous manipulators, pre-processing all commands before acting on them would impose a substantial processing burden on the two DMU computers (the only computers with robot-wide knowledge). This processing would interfere with Ranger's stringent real-time deadlines imposed by the manipulator and joint control algorithms, thereby increasing the possibility of instability. The pre-processing would also introduce substantial additional complexity of the software, increasing both the burden of verification and the likelihood of system failure.[17,18]

Although each LPU and PMU receives commands from both DMUs, no form of active concurrence occurs between the two DMUs, i.e. the two DMUs do not need to agree with each other before sending the LPU or PMU the command. Instead, each LPU and PMU simply takes the "most-safe" of the commands supplied by each DMU. This form of "passive concurrence" reduces the complexity of the communications system and also allows independence of each DMU from the other DMU. Passive concurrence applies to: LPU state commands, PMU state commands, gripper open commands, joint control modes, torque-limits, power-limits, and power relay states.

Of particular concern with safety-critical systems are *common cause failures*.[19] These occur when multiple components fail simultaneously due to the same error, typically as the components are running the same software or are executing on the same hardware. To reduce the potential impact of common cause failures, both functional diversity and data diversity are employed by Ranger in Ref. 19. More than one computer executes each safety check, and, where possible, data from different sources is used by separate computers. Redundant hardware has been used only in the most critical areas (e.g. position sensors), as system-wide application of redundant hardware was deemed unnecessarily expensive and would introduce excessive complexity.

C. System Reaction Time

The control system reaction time is defined to be the time between a failure first occurring and the control system completing transition to the safe state. This system attribute defines how fast the robot can operate, and how close it can approach a non-contactable object. It is a function of the time to detect the failure, the time to communicate the failure to the other computers, the communication time to command all computers to transition to the safe state, and the time to transition to the safe state.

The worst-case system response time is shown in Fig. 6, where the PMU has detected a fault just after frame 0 was transmitted. The intra-robot communications system uses a MIL-STD-1553 infrastructure with a request-response paradigm, whereby the Main DMU queries each computer for their data and the individual computers respond. Due to this delay, when the PMU detects a fault it can not instantly alert the DMUs and must wait for the Main DMU to query it again. This occurs during frame 1, after which the two DMUs notice the PMU-detected failure. During frame 2 the two DMUs determine the resulting course of action, which is to safe the system, and transmit this to all other computers during frame 3. At the end of frame 3 (the start of frame 4) all computers begin executing their respective safing actions. The total communication time, from detection until safing actions start to be executed, is four frames, or 26.67 ms.

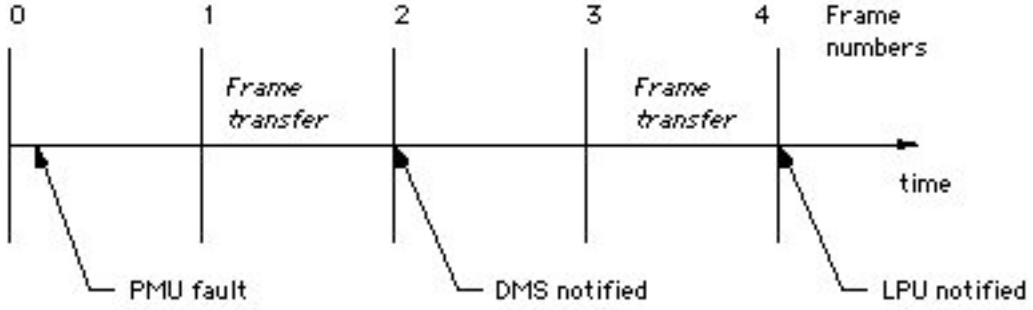

Fig. 6 Worst case system response time, from detection of a PMU failure until all computers have begun transitioning to the safe state.

Each computer has a 7 ms timeout on the communication link; should the link be disrupted the computer will take safing actions once the timeout is reached. In the worst case, if the DMU to LPU link in Fig. 6 was disrupted before frame 3 was transferred, then the LPU would take safing actions 7 ms after the end of frame 2 reception. This amounts to three frames (0, 1 and 2) at 6.66 ms plus the 7 ms timeout, for a total of 3 * 6.66ms + 7 ms = 27 ms. In addition to the actual communication time, the switching time of the power relays adds another 1 ms, and the worst-case LPU or PMU detection time adds another 2 ms (due to the 500 Hz rate). This gives a worst-case system reaction time of 27 ms + 1 ms + 2 ms = 30 ms=0.030 s.

D. Energy Impact Analysis

An energy impact analysis was conducted to determine the closest distance any part of Ranger could approach a non-contactable object.[13] This is the minimum distance at which a failure causing uncommanded motion could be prevented from imparting excessive energy to the non-contactable object, given proper detection of the failure and transition to a safe state. This distance is a function of: the system reaction time, the maximum operating velocity, the inertia seen by the manipulator, and the distance the manipulators will take to drift to a stop once power is removed from the actuators. For the dexterous and video manipulators, the maximum operating velocity is 20 cm/s, yielding a minimum approach distance of 30 cm, while for the slower positioning leg (12 cm/s maximum velocity), the minimum approach distance is 2.5 cm.

The robot's safe operational work envelope can be constructed by concatenating the minimum approach distances of all non-contactable objects. Figure 7 shows a two-dimensional cross-section of the operational envelope, as well as the geometrically simplified work envelope used by the hazard control system. This simplified envelope enables fast distance computations but still contains all the objects to which Ranger has to interface. Note that in some cases the operational envelope appears to go below non-contactable objects, potentially allowing Ranger to approach too close to a non-contactable object. This is considered acceptable as a sufficiently strong contactable object is shielding the non-contactable object; thus the robot would have to penetrate the contactable object before it could damage the non-contactable object underneath. This required modifications only to the payload, and not to the shuttle itself.

V. Autonomous Safety System Validation

Fault tree analysis (FTA)[3] was selected over hazard operability,[2,20] failure mode effects analysis,[11,21] and checklists,[2] as it can determine the events that cause a hazard, fault trees can be reduced to mathematical expressions to determine the hazard probability, and fault trees are easier to construct than, say, event trees, given the small number of hazards relative to the larger number of potential failures. Despite its applicability, FTA does have problems dealing with timing,[22,23] redundancy and differing mission phases,[2] as well concerns regarding the completeness and accuracy of the trees.[2,3]

Ranger's timing model is simple enough to avoid fault trees timing issues, in that a failure is deemed to occur if an item fails to make its deadline. The redundancy of the DMUs and PMUs is handled directly by modeling each instance in the fault trees. LPU redundancies do affect the fault trees, in that a failure at an LPU is not affected by a failure at another LPU, at least in terms of detecting and transitioning to the safe state in the presence of a single fault. Also, requiring additional LPUs to fail in order for a hazard to occur only decreases the hazard probability.[3] The hazard control system does not participate in mission phasing; once the robot is turned on, all safety checks are continually executed.

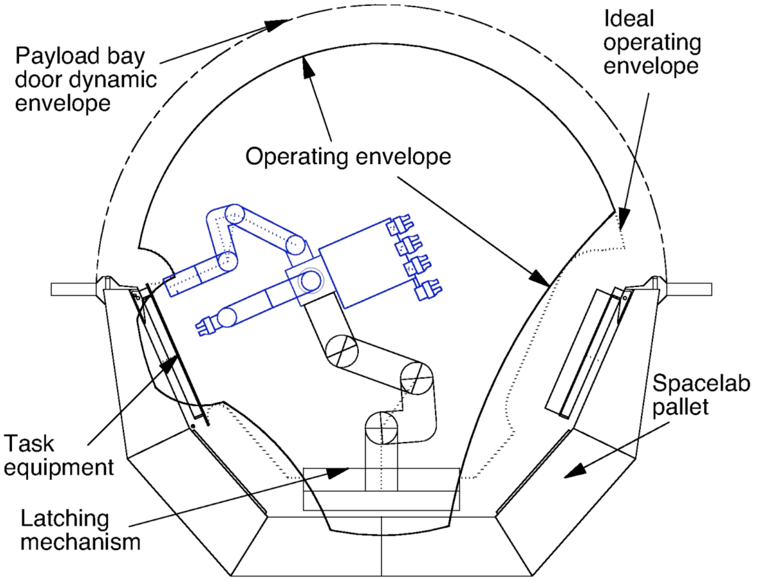

Fig. 7 Dexterous manipulators operating envelope.

Perhaps the most critical concern is that the fault trees are constructed from the results of the PHA. If the PHA fails to identify a hazard then the FTA will not consider this hazard. Also, if some event combination that could cause a hazard is not modeled, then the fault tree, and subsequent qualitative and quantitative evaluations, will be inaccurate. There are no known ways to ameliorate these concerns.

To construct the fault trees, top-level hazards are taken from the preliminary hazard analysis for Ranger TSX. Initiating *first failures* that could potentially lead to a hazard are then determined, and individual subtrees are constructed for each first failure. The fault subtree identifies the combination of *subsequent failures* required, in conjunction with the first failure, to cause the top-level hazard. The analysis does not consider multiple "first failures" that occur simultaneously, nor will the analysis consider situations where the system has continued in the presence of a first failure since these scenarios are outside the fail-safe design specification.

The hazard control system under examination includes all hardware, computers and software within the robot itself. It only involves the operator and control stations in so far as their failure can cause a corrupted or invalid command to be received; their failure cannot contribute to causing a hazard based upon some other failure. The top-level fault tree for Hazard C, "Breaking an object due to excessive torque," is shown in Fig. 8. This fault tree demonstrates that the hazard can be caused by one of four scenarios (C1-C4), each corresponding to a different first failure. The fault tree for Subtree C2, "Force-torque sensor failure causes unknown torque and not safed," is shown in Fig. 9. The four transfer gates, Z4, Z6, Z10, and Z23, all correspond to common subtrees used throughout all three hazards. The remaining Hazard C subtrees, and the trees for Hazards A and B, are not presented here due to space limitations.

All hazards (A-C) have three common subtrees corresponding to first failures *Actuator runaway*, *LPU software failure*, and *Main DMU software failure*. The structure of each type of subtree is generally the same for each hazard, however, the low-level safety checks are specific to the first failure. For example, an actuator runaway failure causes an over-torque situation in Hazard C and consequently the low-level safety check is monitoring for an over-torque situation. For Hazard B, an actuator runaway failure causes release of an ORU and hence the low-level safety check is monitoring for an invalid gripper open.

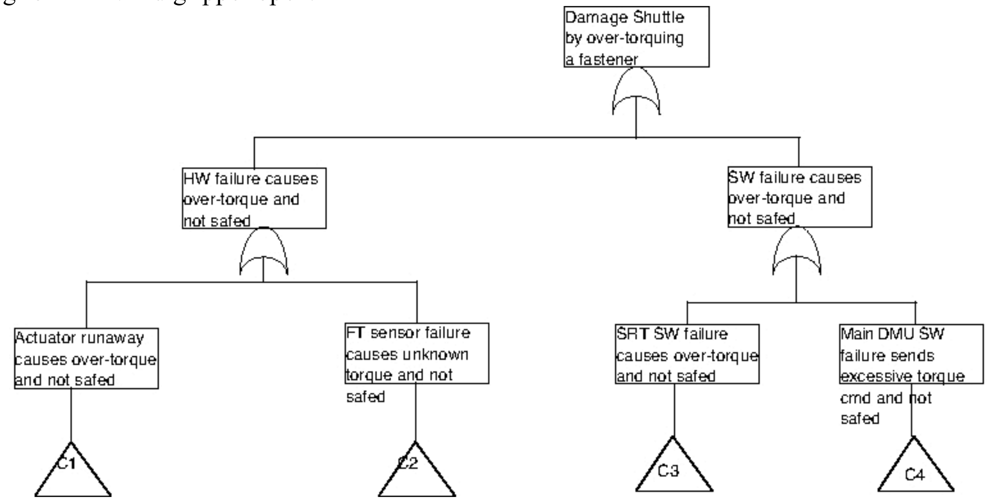
Damage Shuttle by over-torquing a fastener
HW failure causes over-torque and not safed
SW failure causes over-torque and not safed
Actuator runaway causes over-torque and not safed
FT sensor failure causes unknown torque and not safed
SRT SW failure causes over-torque and not safed
Main DMU SW failure sends excessive torque cmd and not safed
C1
C2
C3
C4

Fig. 8 Fault tree for hazard C, "Breaking a fastener due to excessive torque, which damages the Shuttle".

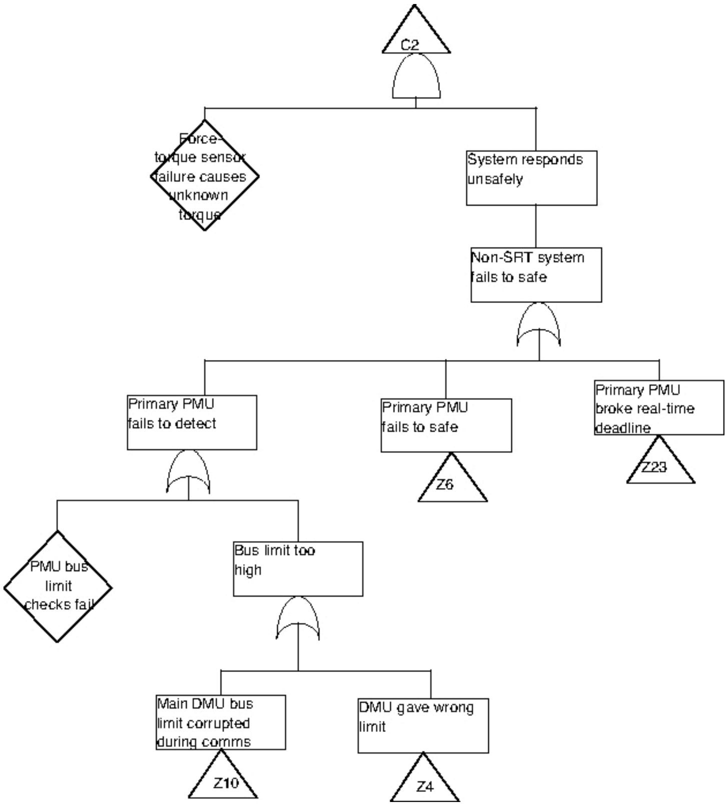
C2
Force-torque sensor failure causes unknown torque
System responds unsafely
Non-SRT system fails to safe
Primary PMU fails to detect
Primary PMU fails to safe
Primary PMU broke real-time deadline
Z6
Z23
PMU bus limit checks fail
Bus limit too high
Main DMU bus limit corrupted during comms
DMU gave wrong limit
Z10
Z4

Fig. 9 Fault tree for "Force-torque sensor failure causes unknown torque and not safed," Subtree C2, a cause of Hazard C.

A. Qualitative Analysis

Once the fault trees have been constructed, each tree's minimal cut sets can be formed. A minimal cut set is defined as "a smallest combination of component failures which, if they all occur, will cause the top level hazard to occur."[3] The minimal cut sets can be ranked by size, providing a qualitative indication of failure importance and the ability to determine if the system meets its design criteria. In this case, no single-component minimal cut sets should exist. If failure events are assumed to be independent, then the failure probabilities associated with minimal cut sets decrease by orders of magnitude as the size of the cut set increases. Hence ranking cut sets gives a gross indication of the importance of the cut set.[3] The distribution of minimal cut sets by size, for each hazard, is shown graphically in Fig. 10. There are over 3500 minimal cut sets in total, with the smallest minimal cut set size 2, the maximum size 13, and the average size about 5.

The data verifies that no single-component cut sets exist, satisfying the design criteria that the system is able to cope with any single failure. Although there exist double component cut sets, the majority of failure scenarios leading to a hazard involve three or more events. Figure 10 also shows that only Hazards A and C have double component cut sets; Hazard B requires at least three failures before a hazard can occur. It can also be seen that there is a significant difference in the number of minimal cut sets for each hazard, largely due to the number of subtrees for each hazard.

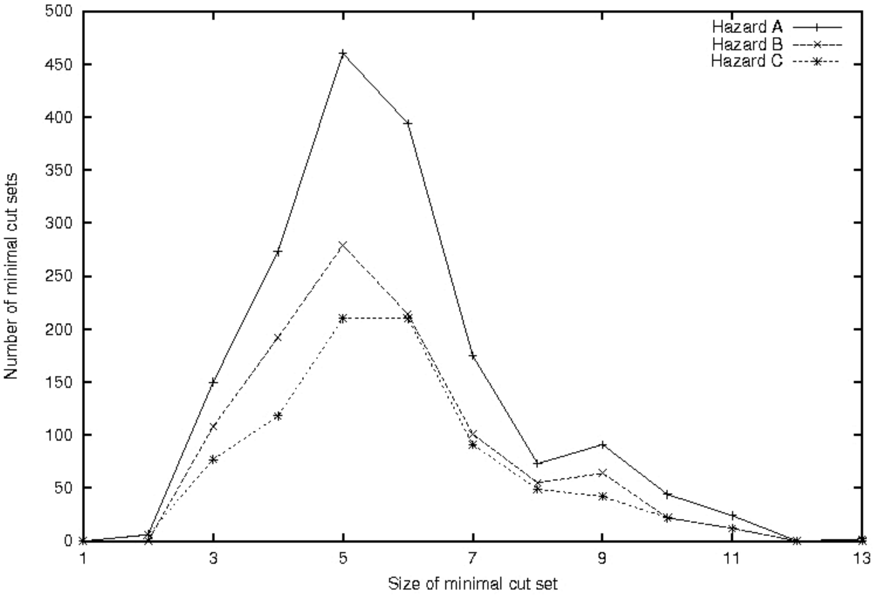

Fig. 10 Distribution of minimal cut set sizes, by hazard.

B. Quantitative Analysis

The fault trees can also be quantitatively evaluated to determine an overall probability of the tree's top-level hazard. An overall probability for each cut set is evaluated based upon the failure probabilities of its constituent events. The top-level hazard is then a function of the probabilities of each of its constituent cut sets.

Quantitative evaluation of a fault tree requires knowledge of whether a failure is repairable or non-repairable.[3] All Ranger TSX failures were considered non-repairable since during a shuttle mission, neither hardware nor software can be repaired or modified. The *lambda-model*[3] was adopted for this analysis and assumes that events are mutually independent and mutually exclusive. The linearized assumptions of this model do result in a conservative estimate of hazard probability. They allow only order of magnitude accuracy, which, given additional imprecision in the input failure probability data, allows only indicative quantitative evaluation.

The failure probability distributions are exponential as the model assumes that the failure probabilities are directly related to component operating times. Hence, the probability $F(t)$ that the component suffers its first failure within time period t, given it is initially working, is

$$F(t) = 1 - e^{-\lambda t} \tag{1}$$

which is accurate to within 5% for $F(t) < 0.1$ [0] and can be approximated to first order by

$$F(t) \sim= \lambda t \tag{2}$$

The derivative of $F(t)$, the probability density function $f(t)$, is

$$f(t) = \lambda e^{-\lambda t} \tag{3}$$

Let $q(t)$ be the component unavailability,

$$q(t) = F(t) \sim= \lambda t \tag{4}$$

the probability that the component is down at time t and unable to operate if called upon.

Let $w(t)$ be the component failure occurrence rate

$$w(t) = f(t) = \lambda e^{-\lambda t} \tag{5}$$

where $w(t)\cdot\Delta t$ is approximately the probability that the component fails between time t and $t+\Delta t$.

For time t small compared to $1/\lambda$, such that $\lambda\, t < 0.1$, $e^{-\lambda t} \sim= 1$, then

$$w(t) = f(t) \sim= \lambda \tag{6}$$

Let $W_i(t)$, the minimal cut set occurrence rate for cut set i, be the probability per unit time of the minimal cut set i failure occurring

$$\begin{array}{rcl} W_i(t) & = & q_2(t)q_3(t)\ \ldots\ q_{ni}(t)w_1(t) \\ & + & q_1(t)q_3(t)\ \ldots\ q_{ni}(t)w_2(t) \\ & & \ldots \\ & + & q_1(t)q_2(t)\ \ldots\ q_{ni-1}(t)w_{ni}(t) \end{array} \tag{7}$$

where n_i is the number of components in cut set i. The first term of $W_i(t)$ is the probability that all components except component 1 are down at time t and then component 1 fails, and similarly for the other terms.

Simplifying $q(t) = \lambda\, t$ and $w(t) = \lambda$, $W_i(t)$ becomes

$$\begin{array}{rcl} W_i(t) & = & (\lambda_2\, t)\ (\lambda_3\, t)\ \ldots\ (\lambda_{ni}\, t)\ \lambda_1 \\ & + & (\lambda_1\, t)\ (\lambda_3\, t)\ \ldots\ (\lambda_{ni}\, t)\ \lambda_2 \\ & & \ldots \\ & + & (\lambda\_{1}\, t)\ (\lambda\_{2}\, t)\ \ldots\ (\lambda_{ni-1}\, t)\ \lambda_{ni} \\ & = & n_i \sum_{i=1}^{ni} (\lambda_i) t^{ni-1} \end{array} \tag{8}$$

$W_s(t)$, the system failure occurrence rate, is the probability per unit time that the top event occurs at time t:

$$W_s(t) = \sum_{i=1}^{N} W_i(t) \tag{9}$$

where N is the number of minimal cut sets. For an operational system such as Ranger, the system failure rate, $W_s(t)$, is the probability of interest.

C. Estimation of failure and hazard probabilities

It is often difficult to obtain good quantitative numbers regarding the safety of a system,[24] primarily due to a lack of historical data. This is the case with Ranger TSX as there is no previous instantiation of this system from which to gather data. Also, software failure rates are difficult to determine as software does not degrade over time as hardware does, nor is the measurement of software failures a mature discipline.[25]

A functionally similar robotic system, the Ranger Neutral Buoyancy Vehicle,[26] has been operational since the mid-1990s and has provided limited subjective failure data. The failure probabilities used here are based on this historical data, with experienced engineering judgment being applied to extrapolate the probabilities to expected Ranger TSX failure probabilities based on the differences between the two vehicles and their development processes. The estimated probabilities of the occurrence of hazards A, B, and C for t=1 hour are 0.0464, 0.00182, and 0.00783, respectively. Given that these are the probability of hazards per hour, the numbers are uncomfortably high. The probabilities of the subtrees that cause the hazards, and their percentage contribution to their parent hazard's probability are shown in Table 1. The top five subtrees stand out as significantly more likely to occur than the remaining subtrees.

Table 1 Estimated probabilities for each subtree and percentage of parent hazard's probability

Subtree	Type	Probability of occurrence in 1 hour	% of parent hazard's probability
A3	Main DMU SW failure causes excess velocity	0.038	81.9
C2	FT sensor failure causes over-torque	0.00775	98.9
A4	Operator failure causes boundary crossing	0.00705	15.2
B1	Operator failure causes gripper open	0.00179	98.2

A2	LPU SW failure causes excess velocity	0.00122	2.6
A7	Main DMU SW failure causes boundary crossing	5.46e-05	0.1
C3	LPU SW failure causes over-torque	5.11e-05	0.7
C4	Main DMU SW failure causes over-torque	3.36e-05	0.4
A1	Actuator runaway causes excess velocity	2.88e-05	0.1
A6	LPU SW failure causes boundary crossing	2.71e-05	0.1
B4	Main DMU SW failure causes gripper open	1.86e-05	1.0
B3	LPU SW failure causes gripper open	1.39e-05	0.8
C1	Actuator runaway causes over-torque	9.15e-07	0.0
A5	Actuator runaway causes boundary crossing	6.36e-07	0.0
B2	Actuator runaway causes gripper open	2.58e-07	0.0

In computing the hazard probabilities for $t = 1$ hour, one assumption of the λ-model was ignored. The λ-model requires $\lambda t < 0.1$ in order to simplify $F(t) = 1\text{-}e^{-\lambda t}$, and the two largest failure probabilities both do not adhere to this. As will be discussed below, operational data to-date indicates that the hazard probabilities are very conservative, and are most likely related to overly conservative failure probabilities.

D. Sensitivity analysis

To identify the variation in hazard probability based on a variation in one input failure probability, a sensitivity analysis was conducted on all three hazards. This analysis varied the failure probability of each first failure and each subsequent failure in that hazard's fault tree, and determined the subsequent variation in the parent hazard's probability. Each failure probability was varied up and down by one order of magnitude. Figure 11 shows the variation in hazard probability as a function of the variation in failure probability, for Hazard C. The plots indicate a general exponentially increasing effect on the hazard probability as a function of increased failure probability. There are significant differences in the amount of increase, as evidenced by the small number of failures that each cause a greater than 100% increase in hazard probability with less than one order of magnitude increase in their failure probability. The graphs for Hazards A and B show the same trends, and are not presented here.

VI. System Performance

Despite the formal cancellation of the Ranger program by NASA, two dexterous arms and a positioning leg have been fully developed for the neutral buoyancy robot. In the course of over 300 hours of Ranger system operations, hardware components have occasionally failed (e.g. encoders, actuators) with predictable and somewhat unspectacular results (e.g. the system stopped). Only three system-wide failures have occurred:

The system failed to detect a damaged joint encoder that subsequently caused a runaway. This occurred during development, when an operator had disabled a safety check. This is not possible in the flight build of the software.

A defect in the system specification caused the control station to incorrectly bound an input parameter. Subsequent hand controller inputs caused a larger amount of motion than expected. This is a case of uncommanded motion caused by a failure in the system specification, rather than being a system runaway. In this case, the autonomous safety system allows the motion to occur so long as the vehicle is operating away from its workspace boundaries. The safety system only takes action when a potential hazard could occur, in this case only when the manipulator violates a workspace boundary.

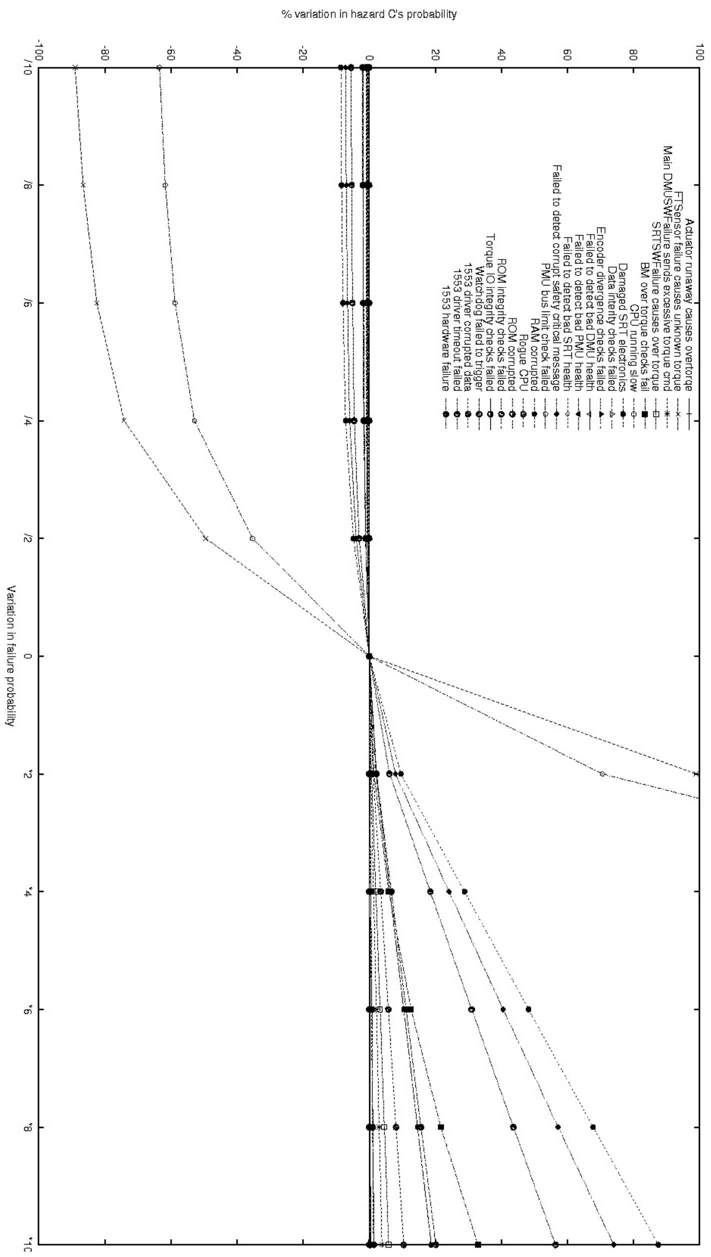

Fig. 11 Percent variation in Hazard A probability as a function of variation in failure probabilities.

Poor power supplied to an LPU held the LPU in reset, preventing it from safing the local motors, resulting in uncommanded motion. This occurred in a development system configuration that did not have working PMUs. In the fully deployed system, both DMU computers would have identified an LPU that was not communicating correctly. In turn, they would have safed the arm, which includes instructing the redundant PMUs to remove power from the arm's motors. This would have prevented the uncommanded motion from occurring. The total time elapsed from the LPU being held in reset to the PMUs removing power is less than 30 milliseconds, a number determined through theoretical analysis and verified through hardware testing.

None of these system failures are related to any of the three system hazards, indeed, two of the failures are possible only during development situations. Given the computed hazard probabilities of 0.0464, 0.00182 and 0.00783 failures per hour, more failures should have occurred in system operations to date. Examination of qualitative data regarding the types of failures seen during Ranger TSX operations, indicate a substantially more reliable system than its predecessor, from which the failure rates have been drawn. As noted previously, obtaining good quantitative numbers is difficult. The extrapolation from a previous system, with a different design and reliability emphasis, appears to have substantially overstated the failure probabilities and hence the hazard probabilities.

The system failures to-date do not indicate a need to reexamine the results of the preliminary hazard analysis. As mentioned above, the second failure is not a hazardous situation, however, the first and third system failures could be directly transformed into subtrees for one or more of the hazards (though still only possible in development situations). For example, system failure three corresponds to an LPU software failure causing boundary crossing as a first failure, while two subsequent failures of the PMUs would be required. There is, however, a need to determine more accurate failure probabilities, or perhaps, to reexamine the structure of the fault trees as they may be producing excessively high hazard probabilities.

VII. Conclusion

This paper has presented the software-based hazard control system for the Ranger Satellite Servicing System. Three hazard types were identified, and a fail-safe approach was implemented to meet the overriding safety requirements. The safety system was implemented in a distributed hard real-time computing architecture, with required reaction times based on artificial workspace boundaries computed from manipulator impact analyses.

The fault tree analysis performed to validate the software safety system yielded mixed results. Qualitative FTA enabled Ranger personnel to verify that the system could handle any single failure, the requirement for the fail-safe hazard control system. This result has also been verified in practice. Despite occasional component failures and operator error, *no* hazard control system failures have occurred since the Ranger system has become fully operational. However, the availability of historical failure data is a key for meaningful quantitative fault tree analysis, as illustrated by the disagreement between the quantitative FTA and actual failures observed during tests. With Ranger, only ad hoc data was available, based on operation of a previous platform that was not comparable in reliability. We believe that the contradiction between the theoretical results and the operational data indicates a problem primarily in the estimation of the failure probabilities, rather than a fault in the design of the hazard control system. Based on this, other design analyses, and the extensive operational history to date, we feel the payload would be safe to fly on the shuttle. Despite this analysis not being required as part of the shuttle payload certification process, NASA would consider all analyses and operational history in determining whether they would certify the payload as safe to fly on the shuttle.

We will continue logging component and system-wide failures as they occur with the goal of better matching the fault trees to observed system performance, enabling increasingly accurate quantitative results over time. Significant neutral buoyancy testing with Ranger is underway to study the tasks and procedures required to robotically service the Hubble Space Telescope. The SSL has also teamed with Woods Hole Oceanographic Institute to develop a fully autonomous single-manipulator undersea vehicle with the ambitious goal of collecting the first biologic and geologic samples of hydrothermal vents under the Arctic ice cap. The manipulator hazard control system will be adapted from the current Ranger system and extended to monitor the autonomy software that will replace current teleoperation systems. The Arctic mission will have very limited time on-station and significant emphasis on successful sample return, resulting in a tradeoff between continued collection of scientific data after minor failures versus use of the pure fail-safe hazard control system implemented on Ranger. Operational history with Ranger and extended FTA will be instrumental in deciding how to relax fail-safe system requirements, since mission success will be gauged both by science data return and safe vehicle recovery.

Acknowledgments

The Ranger Telerobotic Flight Experiment (NCC5-243) was funded by the Space Telerobotics Program of the NASA Headquarters Office of Space Science, Advanced Technology & Mission Studies (Code SM). The authors would like to gratefully acknowledge the entire Ranger development team, without whom this work would not have been possible.

References

[1]Corde Lane, J., Carignan, Craig, and Akin, David, "Time Delay and Communication Bandwidth Limitation on Telerobotic Control," *Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, Boston, MA, May 2000.

[2]Leveson, N. G., *Safeware: System Safety and Computer*, Addison-Wesley, 1995.

[3]Vesely, W. E., *Fault Tree Handbook*, U.S. Nuclear Regulatory Commission, 1981.

[4]Shaw, Roger, "Safety Cases - How Did We Get Here?" *Safety and Reliability of Software-Based Systems, 12th Annual CSR Workshop*, edited by Roger Shaw, Bruges, 12-15 Sept. 1995.

[5]Dunn, W., "Designing Safety-Critical Computer Systems," *Computer*, Nov. 2003, pp. 40-46.

[6]Anderson, T., "Safety - Status and Perspectives," *Proceedings of 12th International Conference on Computer Safety, Reliability and Security*, edited by Janusz Gorski, 27-29 Oct. 1993.

[7]Sommerville, Ian, *Software Engineering*, 5th ed., Addison-Wesley, 1995.

[8]Parnas, D. L., Asmis, G. K., and Madey, J., "Assessment of Safety-Critical Software in Nuclear Power Plants," *Nuclear Safety,* Vol. 32, No. 2, 1991, pp. 189-198.

[9]Potocki de Montalk, J., "Computer Software in Civil Aircraft," *Microprocessors & Microsystems,* Vol. 17, No. 1, 1993, pp. 17-23.

[10]Weber, W., Tondok, H., and Bachmayer, M., "Enhancing Software Safety by Fault Trees: Experiences from an Application to Flight Critical SW," *Proceedings of 22nd International Conference on Computer Safety, Reliability and Security*, 23-26 Sept. 2003.

[11]Ozog, Henry, and Bendixen, Lisa M., "Hazards Identification and Quantification," *Hazard Prevention*, Sept./Oct. 1987, pp. 6-13.

[12]Leveson, N., and Stolzy, J., "Safety Analysis Using Petri Nets," *IEEE Transactions of Software Engineering*, Vol. 13, No. 3, Mar. 1987.

[13]Gefke, Gardell, "RTSX Runaway Impact Energy, Distance and Velocity Analysis," Technical Report DT07-0239, Ver. 0.5, Space Systems Laboratory, Univ. of Maryland, July 1999.

[14]Kandel, Eric R., Schwartz, James H., and Jessell, Thomas M., *Principles of Neural Science,* 3rd ed., Appleton and Lange, 1991.

[15]NASA Space Shuttle Program, "Safety Policy and Requirements for Payloads Using the Space Transportation System," NSTS 1700.7B, May 1996.

[16]NASA Space Shuttle Program Integration, "Computer Control of Hazardous Payloads," MA2-97-083, 1997.

[17]Thane, H., "Safe and Reliable Computer Control Systems," *Proceedings of 16th International Conference on Computer Safety, Reliability and Security*, edited by Peter Daniel, 7-10 Sept. 1997.

[18]Dalcher, D., "Trust, Systems and Accidents: Designing Complex Systems," *Proceedings of 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, 7-10 April 2003.

[19]Storey, N., *Safety-Critical Computer Systems,* Addison-Wesley, 1996.

[20]Stephenson, J., *System Safety 2000*, Van Nostrand Reinhold, 1991.

[21]Hope, S., Bjordal, E. N., Diack, H. M., Eddershaw, B. W., Joanny, L., Ortone, G., Payne, F. G., Searson, A. H., Sedlacek, K. W., and Strien, W., "Methodologies for Hazard Analysis and Risk Assessment in the Petroleum Refining and Storage Industry," *Hazard Prevention*, Jul./Aug. 1983, pp. 24-32.

[22]Gorski, J., Maggott, J., and Wardzinski, A., "Modelling Fault Trees as Petri Nets," *Proceedings of 14th International Conference on Computer Safety, Reliability and Security*, edited by Gerhard Rabe, 11-13 Oct. 1995.

[23]Magott, J. and Skrobanek, P., "A Method of Analysis of Fault Trees with Time Dependancies," *Proceedings of 19th International Conference on Computer Safety, Reliability and Security*, 24-27 Oct. 2000.

[24]Knight, J., and Nakano, L., "Software Test Techniques for System Fault-Tree Analysis," *Proceedings of 16th International Conference on Computer Safety, Reliability and Security*, edited by Peter Daniel, 7-10 Sept. 1997.

[25]Towhidnejad, M., Wallace, D., and Gallo, A., "Fault Tree Analysis for Software Design," *Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop*, 5-6 Dec. 2002.

[26]Parrish, Joseph C., *The Ranger Telerobotic Flight Experiment: A Teleservicing System for On-Orbit Spacecraft*, Telemanipulator and Telepresence Technologies III, Boston, 1996.