4

Designing Safety-Critical Rehabilitation Robots

Stephen Roderick[1] & Craig Carignan[2,1]
[1]*University of Maryland &* [2]*Georgetown University*
U.S.A.

1. Introduction

In recent years, robots have made substantial in-roads in the medical field and are gradually finding their way into clinical practice. Intuitive Surgical's *da Vinci® surgical robot* broke ground in 1998 by performing the first tele-robotic surgery to repair a heart valve (Salisbury, 1998; Guthart & Salisbury, 2000). Accuray's *CyberKnife* radiotherapy robot began treating head, neck and upper spine tumors in 1999 by combining image guidance with a robotically-directed radiation beam (Adler et al., 1997). In 2002, Interactive Motion Technology began therapy of stroke patients with the *InMotion*[2] robot, also known as the MIT-Manus (Krebs et al., 2002). These devices and many others under development have provided researchers and doctors alike with capabilities not previously available.

These additional capabilities, however, have also brought with them the issue of safety – these are *safety critical systems* in which a single malfunction can endanger the life of the patient. In contrast with traditional robotic systems, medical robots must enforce the safety of the patient as an object within its workspace, while also being able to treat the patient. This dichotomy creates the need for a *safety system* that can allow the robot to interact with the patient, while also enforcing all necessary safety precautions.

Human fatalities resulting from medical treatment with machines is unfortunately all-to-real. The *Therac 25*, a radiation therapy machine developed by the Atomic Energy Commission of Canada, was involved in six known accidents between 1984 and 1987. Five patients died as a result of massive overdoses of radiation when a high power electron beam was activated without the target tumor having been rotated into place (Leveson & Turner, 1993). Had the machine's software detected the fault, the accident would have been averted. Radiation therapy machines are now required to have hardware interlocks to prevent activation of the high-energy electron-beam unless the target is in place.

A similar tragedy occurred at the National Oncology Institute in Panama during 2000 and 2001. Twenty-eight patients were overexposed during radiation therapy for cancer treatment, after use of a computerized treatment planning system. Dosage calculations from the planning system had errors of up to 105%. By August 2005, 23 out of 28 overposed patients had died, of which at least 18 were attributed to radiation effects (Borras, 2006).

Dangers in medical robotics are not confined to surgical systems. In powered orthoses or "exoskeletons" being developed for rehabilitation, humans are basically encapsulated in the device creating a potentially hazardous situation. Powered leg exoskeletons such as the the *Lokomat™ Gait Orthosis* are being used to train stroke and spinal cord injury patients how to

Source: Rehabilitation Robotics, Book edited by Sashi S Kommu,
ISBN 978-3-902613-04-2, pp.648, August 2007, Itech Education and Publishing, Vienna, Austria

walk again (Bernhardt et al., 2005). Arm exoskeletons such as *ARMOR* (Mayr et al., 2006), ARMin (Nef et al., 2006), RUPERT (He et al., 2005), and *UW Prototype III* (Rosen, 2005) are being developed for therapy of hemiparetic stroke patients. The MGA Exoskeleton is being used for shoulder rehabilitation and is comparable in strength to the average adult male (Carignan et al., 2005). While all of these systems have adequate sensors to control the robot, they are insufficient to enforce patient safety.

It is important to realize that safety is not an absolute concept – a system can only be built to reduce the risk of an accident to an acceptable level (Shaw, 1995; Dunn, 2003). Safety is also an attribute of the entire system and is not driven by only certain components of the system. This requires that safety analyses include all system components: software, hardware, and the operators (Anderson, 1993; Leveson, 1995; Sommerville, 1995).

This chapter will detail a system safety design process that can systematically evaluate the design of a rehabilitation robot against its project safety and failure rate criteria. When these criteria are not satisified, the process can identify system components or failure combinations that require additional design consideration, so that the project criteria might be satisfied. Modifications directed by this process can result in a sufficiently safe system design for safety-critical rehabilitation robot applications. The application of this process to two case studies will be presented. The first case study is the MGA Exoskeleton introduced above, which will be used as an illustrative example. The second case study is a multi-arm space robot experiment, to which this process was originally applied. This robot has several years of operational history against which this process can be evaluated.

Definitions

There are many different definitions of safety-related terms. For consistency, the definitions used in this chapter are taken from Vesely (1981) and Leveson (1995):

- A *failure* is an abnormal occurrence
- A *fault* is a higher-order event caused by one or more failures.
- A *hazard* is a system state and other environmental conditions that inevitably leads to an accident.
- An *accident* is an undesired and unplanned event that results in a level of loss, in this case, injury to the patient.
- *Safety* is freedom from accidents.

Fig. 1 depicts symbols used in this work to represent fault events and gates.

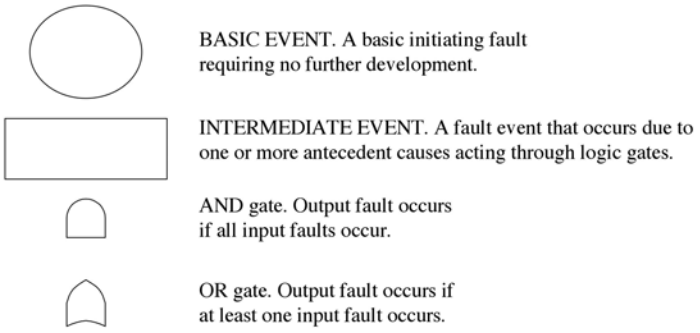

Fig. 1. Symbols used in fault trees.

2. Safety analysis techniques

Previous medical robotics have had to address the issue of patient safety (Varley, 1999; Taylor et al., 1991, Duchemin et al., 2004). One of the unique aspects of rehabilitation robots is that the human in-the-loop is the patient. With surgical or radiological systems such as *Da Vinci* and *Cyberknife*, a patient is being "operated" on by the robot; however, a clinician is directing the robot. With a rehabilitation robot, the subject may fill the role of both patient and operator. This introduces additional safety criteria over and above that used in more traditional medical robots.

Unfortunately, there is no industry-standard approach to designing these safety-critical robot systems. Despite this, numerous other fields have standard and accepted analytical methods used to design safety-critical systems (Weber et al., 2003). These methods come under the banner of "system safety engineering" (Stephenson, 1991; Blanchard, 1991), and have been used to develop safety-critical systems in domains ranging from aircraft flight management systems (Parnas et al., 1991) to nuclear power plants (Potocki de Montalk, 1991).

2.1 Current approaches

Some of the more common design techniques are described in the following sections and can be loosely categorized as either "top-down" or "bottom-up" techniques. Top-down techniques typically work from a high-level description, and attempt to identify combinations or sequences of components that contribute to system level events. Bottom-up techniques typically work from the component level outward and upward. Hybrid techniques combine both top-down and bottom-up approaches. A brief description is now given of the more standard techniques.

2.1.1 Top-down approaches

Preliminary hazard analysis (PHA) is an early phase of lifecycle-based hazard analysis, which also involves system and subsystem hazard analysis (Leveson, 1995). Its objective is to assess potential hazards caused by the system, to identify inherent hazards, and to assess the criticality of arising accidents (Vesely, 1981). The resultant hazards can be used as inputs to fault tree analysis or similar techniques.

Hazard operability is a qualitative, creative thinking technique developed by, and primarily used by, the chemical industry. It is simple, but very labor intensive, and is designed to identify and analyze hazards systematically (Stephenson 1991; Leveson, 1995). As with PHA, the results of a hazard operability study can be used as inputs to fault tree analysis or similar techniques.

Fault tree analysis (FTA) is a deductive technique to determine the sequence of faults that could cause a given top-level hazard (or *top event*). The resulting fault trees can be represented as boolean expressions and reduced to the minimum combination of failures that could cause the top-level hazard. The trees can also be quantitatively evaluated to provide estimates of the probability of the top event occurring, given the probabilities of the constituent failures. Fault trees can have trouble dealing with timing (Gorski et al., 1995), redundancy, and differing mission phases (Leveson, 1995). It is also a time-consuming, qualitative technique, although its absolute accuracy is usually secondary to identifying failure sequences (Ozog & Bendixen, 1987). The determination of the top-level event is critical; failure to determine a top-level event by a PHA, for example, results in the hazard not being examined and, consequently, the system may not cope with that hazard (Leveson, 1995; Vesely, 1981).

While typically applied to a system design, FTA has recently been applied to software to determine how failures in its implementation could cause a hazard (Leveson 1984; Knight & Nakano, 1997). It can be used to determine cases where the implemented design could cause a hazard, or show which modules are most critical to the safe operation of the system. Methods exist to potentially build the trees from a logical system model (Bruns & Anderson, 1993), or from a completed code base (Voas, 1995), though this may occur too late in the design cycle to be cost-efficient. It also difficult to evaluate such fault trees quantitatively, as there exist few methods to assign probabilities to failure of software logic.

2.1.2 Bottom-up approaches

Failure mode effects analysis (FMEA) is traditionally used to predict equipment reliability and emphasizes successful functioning of a component as opposed to the failure of the component. It is a systematic technique that is system-oriented, not hazard-oriented (Hope et al., 1983). It is more time-consuming than fault tree analysis and does not cope with multiple failures, timing, or redundancy (Ozog & Bendixen, 1987; Leveson, 1995).

Failure modes, effects, and criticality analysis (FMECA) is a very similar technique, that adds extra steps and data to an FMEA related to controls and control procedures.

Event tree analysis is an inductive method to identify outcomes of a given initiating event and can identify the components that most contribute to a failure. It is practical for independent events and a stable event chronology, and may provide top events for subsequent fault tree analysis. Event trees can become very large and suffer difficulties with timing data and more than two states. They may also require fault tree analysis to develop failure probabilities for branches, and analysts have to be able to define all initiating events (Ozog and Bendixen, 1987; Leveson, 1995).

2.1.3 Hybrid approaches

Cause-consequence techniques combine fault-trees and event-trees. They work by selecting a critical event and determining the contributing factors using fault tree analysis and the resulting consequences using event tree analysis. They can represent delays and event combinations, but they can become large and unwieldy. In addition, their outcome is only related to the cause being analyzed (Hope et al., 1983; Leveson, 1995).

2.1.4 Synthesized techniques

Applied in isolation, none of the above techniques can produce a system design that is sufficiently safe. Synthesizing combinations of these techniques can produce a system safety design process that utilizes the strengths of each technique, thereby providing a design methodology that not only identifies hazards and their contributing fault scenarios, but can also potentially evaluate the design qualititively and quantitatively. This synthesized technique may be used to determine the specific need of additional system components, required to enforce the project's safety critiera.

2.2 Previous synthesized safety analyses

Lankenau et al. (1998) combined FTA along with formal methods, to develop a safety layer for an autonomous wheelchair. They also applied a model checker over the fault trees, to attempt to verify that the system could never fail.

Combining FTA with an unspecified failure mode analysis technique, Cavallaro & Walker (1997) evaluated the safety and reliability of a manipulator system for hazardous material retrieval. They show results for only one hazard, and noted that some failure modes were not considered due to lack of details on software configuration and the operator interfaces.
Guiochet & Vilchis (2002) combined FMEA and FTA in a safety analysis of the design of an ultrasound robot for tele-echography. The two analyses were used in conjunction due to the complementary forward/backward approaches. While a hazard analysis is mentioned as part of their process, it is unclear where it fits into their process. They use FMEA to identify the failure leading to an accident, and use that failure as a top event for subsequent FTA. As some accidents require multiple failures either combinatorially or sequentially, these hazards will not be identified by the FMEA. However, results of the FMEA could be used to identify corrective measures for the system.

2.3 Chosen synthesized technique

For this work, a synthesized approach was chosen that combines PHA and FTA. This provides the means to enumerate the hazards a system presents, a method to determine the fault sequence and/or combinations that can cause the hazards, and both qualitative and quantitative metrics against which project safety and failure rate criteria can be compared. FTA was chosen over such techniques as FMEA, event trees or cause consequence trees since, in our experience, there are far fewer hazards than there are failure combinations leading to such hazards. Construction of fault trees requires less work than the other techniques in this scenario — however, the issues of timing and redundancy in a robot system must be explicitly addressed within the FTA.
For the two case studies presented here, the distributed, hard real-time nature of the robot is leveraged alongside safety checks to specifically target components that do not make their deadlines. It is considered a failure if a computing component does not make an internal processing deadline, or an external communication deadline. This is an FTA primitive event, allowing timing-related failures to be explicitly modelled in the fault trees.
Redundant components may be explicitly modelled within the fault trees. This leads to more work in the fault tree construction but ensures that the redundancy is dealt with directly. It is also possible to ignore redundancy under certain circumstances, where the modelling of additional components would only lead to decreased hazard probabilities. In this case, the resultant hazard probabilities will err on the conservative side.
The approach described here was initially developed for a dexterous robot designed to fly on NASA's space shuttle (Roderick et al., 2004). This system was the first (and to our knowledge, the only) American robotic system to be certified through three of the four phases of the NASA Space Shuttle Safety Review process. This pioneered a solely computer-based hazard control system for payloads operating on the shuttle, and demonstrated successful application of this technique to a safety-critical robot system.

3. System safety design process

The overall goal of this process is to evaluate a system design to determine whether it is "sufficiently safe". The concept of sufficiently safe is one that the specific project must establish, which forms the basis for the *project's safety criteria*. This will include factors such as regulatory standards as well as the potential consequences to the patient of an accident occurring. It is not possible to make a system absolutely safe; however, if the likelihood of

an accident is small enough — or the consequences of an accident are negligible enough — the system may be considered sufficiently safe (Anderson, 1993; Shaw, 1995). At some point, continuing to modify a system design to cope with ever more incredible failures simply results in an excessively complex design, and a subsequent reduction in overall system reliability and/or safety (Duchemin et al., 2004). Notably, this concept of accepting some degree of risk is enshrined in UK law under the *principle of reducing risks as low as reasonably practicable* (McDermid, 2001).

3.1 Process description

A flow diagram illustrating the system safety design process is shown in Fig. 2. The basic inputs to this process are an initial system design description, the project safety criteria, and the project failure rate criteria. During a given iteration of the system design, the specific deliverables or outputs from this process are a list of hazards from the PHA, the fault trees from the FTA, and any qualitative and/or quantitative results from the FTA. These outputs also form the overall outputs of the process.

The initial system design is evaluated through a PHA, and the hazards identified by the PHA each constitute top events from which FTA can begin. Each top event is considered individually, and the immediate, necessary, and sufficient causes by which this event could occur are identified. These immediate events are summarily examined for their causal events, and this step-by-step analysis continues until individual component failures are reached. These component failures are the basic causes that, when combined in the manner indicated by the fault tree, guarantee that the top level hazard will occur. (See Leveson (1995) for further details of PHA, and Vesely (1981) for fault trees and their construction.)

The resulting fault trees can be qualitatively examined to determine if the project's safety criteria are being met, i.e., whether the system is sufficiently safe. If not, additional components may be added in an effort to deal with the specific safety issues raised by the FTA. The system design is then modified accordingly, and the process begins again. Once the FTA results show that the project's safety criteria are met, the system design can be considered sufficiently safe.

The qualitative analysis entails forming and evaluating the minimal cut sets for each fault tree. 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* (Vesely, 1981). 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. If failure events are assumed to be independent, then the failure probabilities associated with minimal cut sets can decrease by several 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 (Vesely, 1981).

Once the system design has satisfied the project's safety criteria, the fault trees can be quantitatively analyzed to determine an indicative system failure rate. 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. If the computed system failure rate is higher than desired, then sensitivity analysis of the input fault probabilities can indicate which components signficantly contribute to the overall rate. These specific components can be flagged for additional inspection or higher-tolerance part replacement, in an effort to reduce their failure probability. In addition, further iterations of the safety design process may occur with the aim of modifying the system design to reduce the system failure rate, while still

satisfying the project's safety criteria. A more detailed description of the quantitative analysis is outlined in the next section.

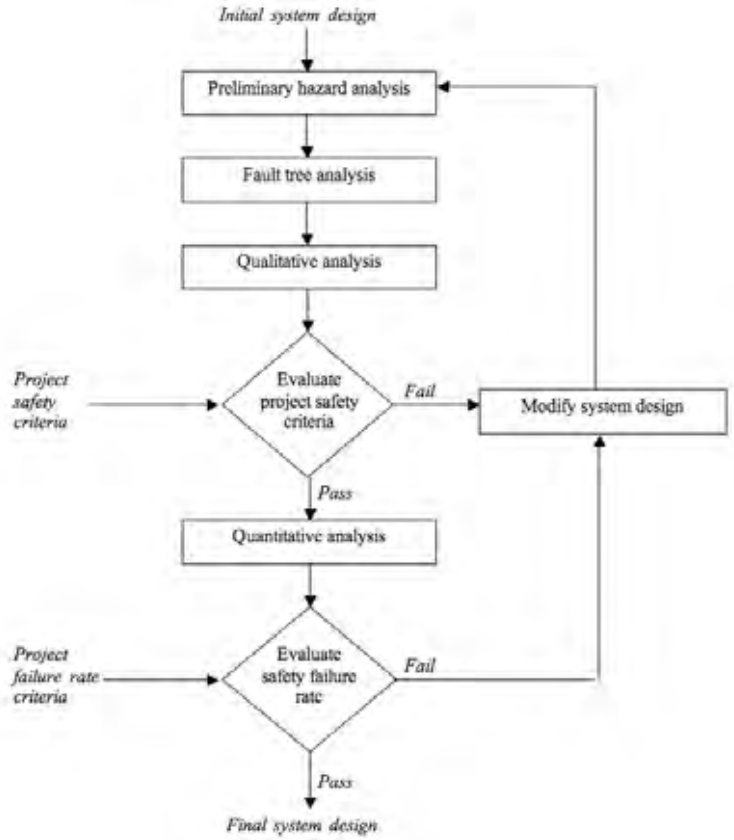

Fig. 2. Approach to system design for safety.

3.2 Quantitative Analysis

The *lambda-model* adopted for this analysis assumes that events are mutually independent and mutually exclusive, and requires knowledge of whether a failure is repairable or non-repairable (Vesely, 1981). The linear assumptions of this model will result in a conservative estimate of hazard probability, thus allowing only an order of magnitude accuracy. Given the additional imprecision in the input failure probability data, this allows only indicative quantitative evaluation. The following development is summarized from (Vesely, 1981).

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 $F(t) < 0.1$ and can be approximated to first order by

$$F(t) \cong \lambda t \tag{2}$$

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,

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

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 the approximate probability that the component fails between time t and $t+\Delta t$. For time t small compared to $1/\lambda$, such that $\lambda t < 0.1$, $e^{-\lambda t} \cong 1$,

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

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

$$\begin{aligned} W_i(t) \;=\; & q_2(t)q_3(t) \ldots q_{n_i}(t)w_1(t) \\ +\; & q_1(t)q_3(t) \ldots q_{n_i}(t)w_2(t) \\ & \ldots \\ +\; & q_1(t)q_2(t) \ldots q_{n_{i-1}}(t)w_{n_i}(t) \end{aligned} \tag{7}$$

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. Substituting (4) and (6) into (7), $W_i(t)$ becomes

$$\begin{aligned} W_i(t) \;=\; & (\lambda_2 t)(\lambda_3 t) \ldots (\lambda_{n_i} t)\lambda_1 \\ +\; & (\lambda_1 t)(\lambda_3 t) \ldots (\lambda_{n_i} t)\lambda_2 \\ & \ldots \\ +\; & (\lambda_1 t)(\lambda_2 t) \ldots (\lambda_{n_{i-1}} t)\lambda_{n_i} \\ =\; & n_i \Sigma_{i=1,n_i} (\lambda_i) t^{n_i - 1} \end{aligned} \tag{8}$$

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

$$W_s(t) = \Sigma_{i=1,N} W_i(t) \tag{9}$$

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

4. Case Study: The MGA Exoskeleton rehabilitation robot

The MGA Exoskeleton mentioned briefly in the introduction will be used as an illustrative example of how to conduct this design process. This arm exoskeleton was designed to treat shoulder pathologies such as rotator cuff tear and shoulder impingement syndrome. It has four shoulder degrees of freedom (three rotations and a scapula elevation), an elbow pitch joint, and a passive forearm pro/supination joint. It is capable of producing 134 N-m of torque at each shoulder joint and 64 N-m at the elbow, which is comparable to the output of the average adult male (Caldwell et al., 1998). The human interfaces consist of a torso mount for the scapula, an orthotic splint for the upper arm, and a hand grip. The system is fast and powerful, and because of the arm restraints, fast patient egress is not possible.

The project safety criteria specifies that no single failure can cause a hazard and the the system must be *fail-safe*. A fail-safe system is one that will achieve a safe state in the presence of a detected fault (Dunn, 2003; Roderick et al., 2004). When a fault is detected, the exoskeleton should either a) halt arm motion and hold the current position, or b) safe the arm by removing power to the motors. It is important to note here that fail-safe does not necessarily mean powering off the robot – that might actually endanger the patient more. For example, placing the exoskeleton in a passive, gravity-assist configuration might be the primary fail-safe state.

Conducting a fault tree analysis is a top-down process in which the operating modes and control system form an integral part. The procedure, illustrated in Fig. 3, begins with the task protocol such as enabling a resistive rotational movement about the shoulder. The task then determines the operating mode of the robot. For example, performing a shoulder abduction exercise would require that the resistance profile about a particular shoulder axis be controlled. The operating mode then determines what controller(s) needs to be activated to realize that particular protocol. Finally, the safety system must monitor and protect the patient from potential hazards during execution of the task.

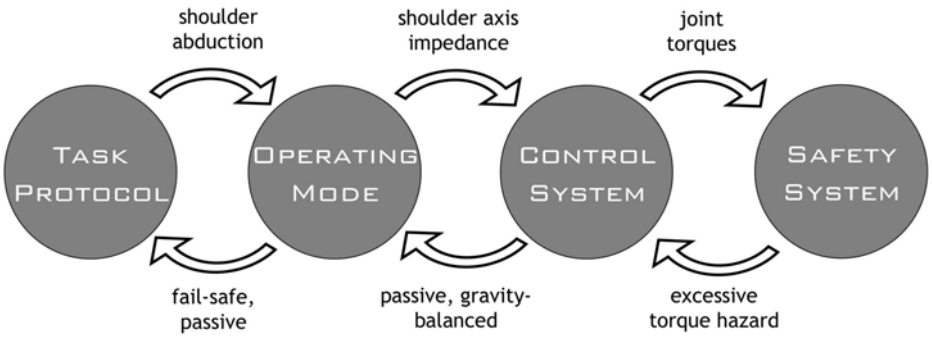

Fig. 3. The medical robot design begins with the task and loops to the safety system and back again.

4.1 Protocols and Operating Modes

There are basically two classes of shoulder therapy protocols currently being implemented on the exoskeleton: *iso-lateral exercise* and *functional training*. Iso-lateral exercises are those that occur around a single rotation axis of the shoulder or along a straight line path of the hand. Functional training involves more general movement of the hand through three-dimensional space which occurs in everyday tasks.

Iso-lateral exercises closely mimic those currently performed manually or with the assistance of exercise machines. Examples of shoulder rotation exercises include shoulder abduction/adduction and internal/external rotation as shown in Fig. 4 (Liszka, 2006). In rotational exercises, the motion of the shoulder joints is determined by the resistance about the desired shoulder axis of rotation. Examples of exercises involving straight line motion of the hand include upright rows and wall push-ups.

Fig. 4. Exoskeleton shown at (a) full shoulder adduction (b) 90 degree shoulder abduction, (c) mid-elbow flexion, and (d) near full lateral rotation.

During *functional training*, the patient views the simulated task and a representation of their arm through a head mounted display, while the exoskeleton provides haptic feedback to the patient. A force sensor located at the hand gripper senses the forces being exerted by the patient during "contact" with the virtual environment and relays them to the controller which commands the exoskeleton in response to the interaction. Examples of functional training are proprioceptive neuromuscular facilitation (PNF) patterns and simulations of activites of daily living.

4.2 Control System

The modular control architecture implemented on this system is shown in Fig. 5. The exercise protocol is first parsed into a control mode based on the desired activation of the arm joints. This code then determines which controller(s) should be activated for the possible combinations of arm groups: scapula, shoulder, elbow pitch, and elbow orbit. These groups can implement impedance (torque command) and admittance (position command) modes depending upon the availability of force sensing and the impedance settings. In the case where both modes are feasible, e.g. rowing, the level of impedance often determines which mode will be implemented.

As an example, the impedance control module used for controlling the resistance about an arbitrary shoulder axis is shown in Fig. 6. The stiffness and damping about the desired Cartesian axes of rotation are set using the desired impedance Z_d. The desired orientation of

the glenhumeral (GH) joint is input to the controller and "differenced" with the sensed GH orientation computed from the forward kinematics of the shoulder joints to produce the angle-axis error. The GH angular velocity error is then multiplied by the desired impedance to produce the desired GH torque. This torque is then mapped to joint torques τ_d via the transpose-Jacobian, which is added to a feedforward compensation torque to produce the torque servo command τ_d.

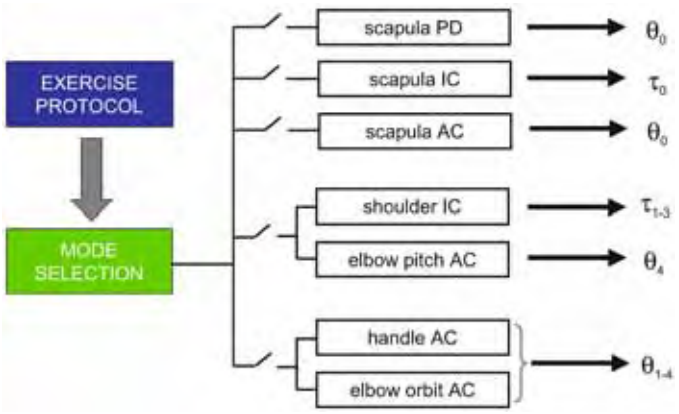

Fig. 5. Modular control architecture for the MGA Exoskeleton.

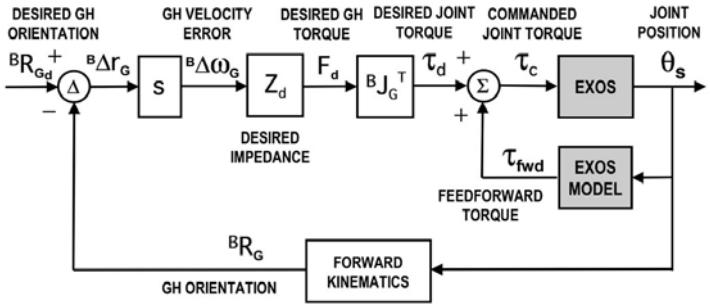

Fig. 6. Exoskeleton impedance controller for shoulder axis rotation.

4.3 Safety System

The above control modes and their associated control systems define the minimum suite of sensors and actuators that are required to carry out operations, a partial diagram of which is shown in Fig. 7. This constitutes the initial system design description, and must now be evaluated to determine whether it satisfies the safety criteria of the project.

The PHA for this project involved defining the system boundary, determining the types of possible accidents, and identifying hazards that may cause such accidents. The following were used to aid in identification of the hazards: lessons learned from previous robotic systems, historical operational data, and critical examination of the results of the PHA conducted on a previous robot with a similar architecture (the second case study in this work). Conducting a PHA is an iterative engineering task, and required several sessions.

Given the initial system design description (shown visually in Fig. 7), the PHA identified three potential hazards (in this context, "excessive" means an unhealthy level, leading to injury):

Hazard A: Moving the patient outside their safe position range.
Hazard B: Moving the patient at an excessive velocity.
Hazard C: Applying excessive torque to the patient or, conversely, allowing the patient to apply excessive torque against the robot.

Each of these hazards will be considered in turn, through the FTA and subsequent modifications to the system design.

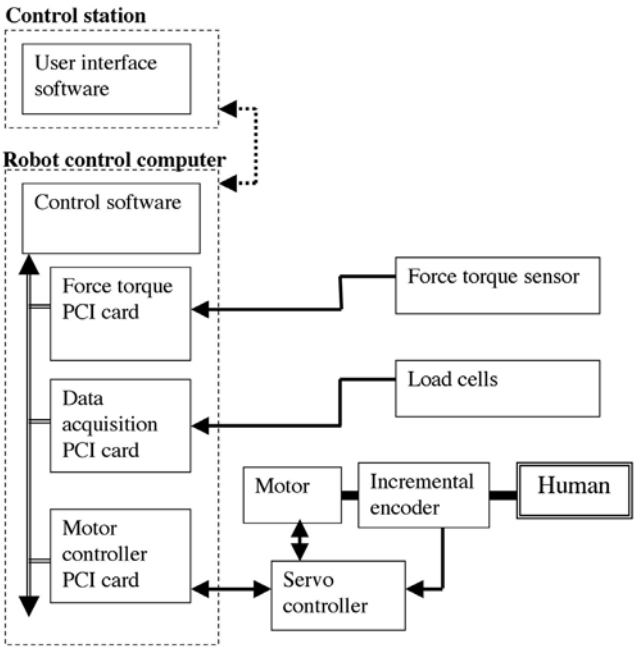

Fig. 7. Initial system design. This is the minimum suite of sensors and actuators required for operations. For clarity, only major system components are shown. Also, the set of sensors and actuators required for only one of the five degrees of freedom is shown.

4.3.1 Hazard A: Moving the patient outside their safe position range

A fault tree developed from the initial system design of Fig. 7, and the top event "Moving the patient outside their safe position range", is shown in Fig. 8. The top event can be caused by any one of numerous possible intermediate events, due to the OR gate attached to the top event (see definitions in Fig. 1). The intermediate event shown, "Uncommanded motion due to joint runaway", can be caused solely by a failure of the incremental encoder, which is a primary component of the control law used to drive the motor.

This scenario fails the project safety criteria, and so additional components were added to the system and the PHA and FTA were repeated. The modified system design is shown in Fig. 9, where the shaded components, an absolute encoder and a power amplifier, are additions over the initial system design (Roderick & Carignan, 2005).

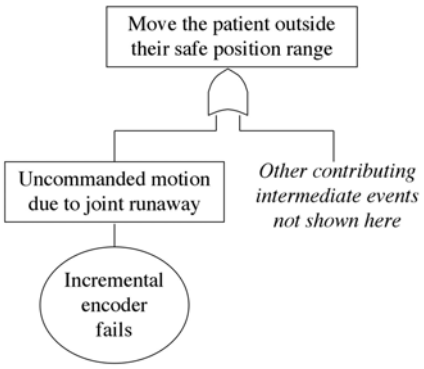

Fig. 8. Fault tree for the initial system design and the top event "Moving the patient outside their safe position range". This fault tree shows that a single fault, that of the incremental encoder, could cause the top event to occur.

The fault tree for this top event and the modified system design is shown in Fig. 10. This fault tree considers the addition of a second encoder and a software-based divergence check to the system design. The divergence check is designed to detect a failed encoder by comparing the values of the two encoders, and flagging a fault if they differ by more than a prescribed tolerance. This fault tree demonstrates that the addition of the second encoder and the encoder divergence check will satisfy the project safety criteria for this hazard: no one failure is capable of producing the hazard.

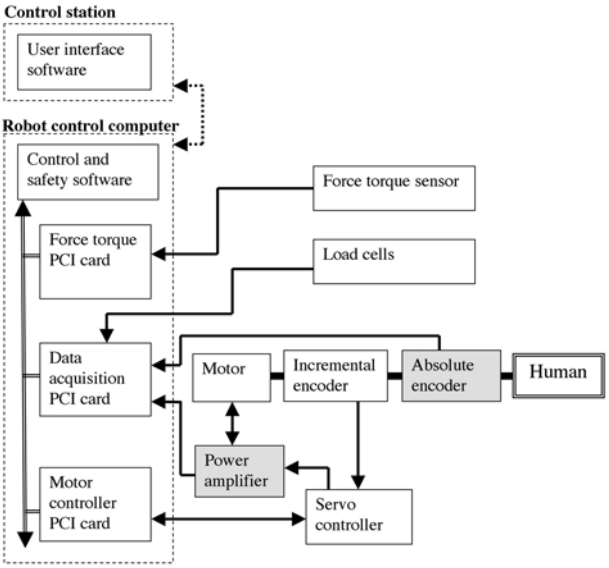

Fig. 9. Modified system design with additional components to satisfy project safety criteria. The additions over the initial system design of Fig. 7 are shaded. For clarity, only major system components of relevance are shown. Also, the set of sensors and actuators required for only one of the five degrees of freedom is shown.

While the modified system design does prevent a single failure from causing this hazard, closer examination of Fig. 10 shows that a double failure could still cause the hazard. If both encoders fail in such a way that they output almost the identical same value they would pass the encoder divergence check. While this failure combination is possible, particularly for certain values (depending on the encoders construction, 0 or -1 are likely candidates), it is highly unlikely to occur at the same time, and thus could be deemed an "incredible" failure and removed from further analysis. While further modifications to the system design, such as a third encoder, may enable detection of such situations, the additional system complexity may be unwarranted as well as potentially contributing to lower system reliability. The tradeoff between these measures is beyond the scope of this chapter.

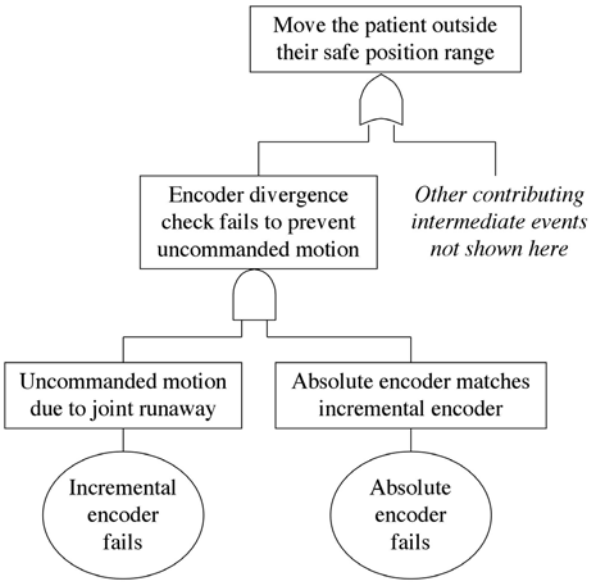

Fig. 10. Fault tree for the modified system design and the top event "Moving the patient outside their safe position range". This fault tree indicates that two simultaneous faults are required for the intermediate event shown to cause the top event to occur.

To help determine the overall likelihood of such incredible failures occurring, the fault trees may be quantitatively evaluated. As noted in Section 3.1, FTA is generally a qualitative technique, who's quantitative accuracy is indicative at best. Quantitative analysis may therefore simply be beneficial in ranking failures by probabilistic likelihood, versus using the output probabilities as absolute indications of safety (Roderick, 2000).

4.3.2 Hazard B: Moving the patient at an excessive velocity

The fault trees for this hazard are very similar in structure to those for the previous hazard. This is primarily due to the system computing velocity based on sequential encoder readings, and hence there are identical measures to sense excessive velocity or to detect a failed component that contributes to velocity sensing. Thus, this hazard is not considered any further here.

4.3.3 Hazard C: Applying excessive torque to the patient

A fault tree for the initial system design and the top event, "Applying excessive torque to the patient", is shown in Fig. 11. A single failure of the servo controller, which is responsible for providing power to the motor, is capable of producing uncommanded motion and hence, potentially, applying excessive torque to the patient. The fault tree of Fig. 12 is for the modified system design and shows the addition of a separate power amplifier with built-in motor current sensor, as well as a software-based motor power check (not shown). This power check compares the motor current draw with the requested output of the servo controller, to determine if either component is at fault. This fault tree indicates that the project safety criteria are satisfied by these additions.

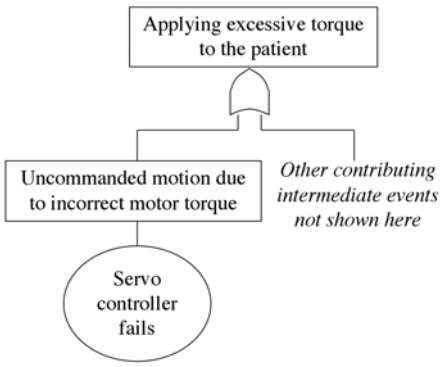

Fig. 11. Fault tree for the initial system design and the top event "Applying excessive torque to the patient". This fault tree shows that a single fault, that of the servo controller, could cause the top event to occur.

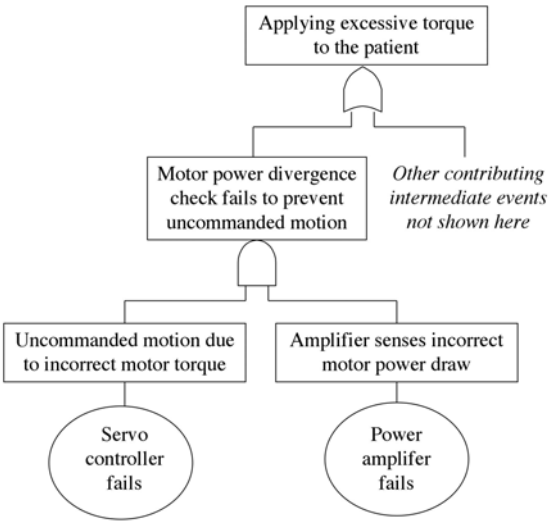

Fig. 12. Fault tree for the modified system design and the top event "Applying excessive torque to the patient". This fault tree indicates that two simultaneous faults are required for the intermediate event shown to cause the top event to occur.

4.4 Summary

Through several iterations of system safety engineering, the final system design of Fig. 13 was reached. Comparison with the initial system design of Fig. 7 shows the addition of extra sensors for each degree of freedom, as well as multiple emergency stop capabalities. Note that for brevity of explanation, only trivial examples of fault trees and their associated system design modifications have been shown. This project is also a work-in-progress, so little operational data exists at this time. Therefore, operational results from a similar robotic system with a substantial amount of operating time will now be considered.

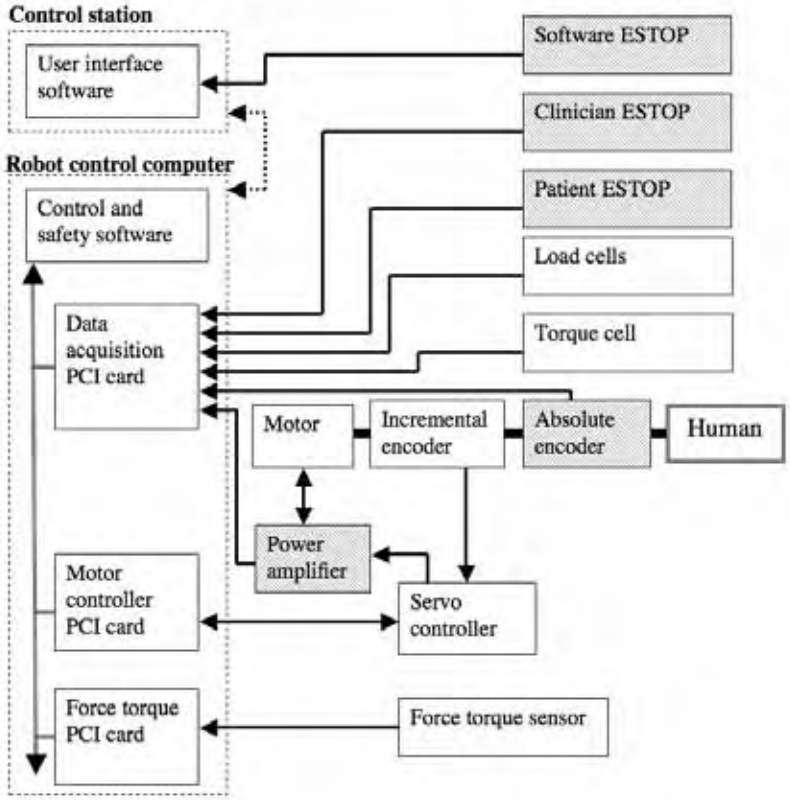

Fig. 13. Final system design showing the additional safety-related components (shaded), over those required solely to realize the system's control modes.

5. Case study: The Ranger Telerobotic Shuttle Experiment

The system safety design process described here was originally applied to the Ranger Telerobotic Shuttle Experiment (RTSX). This four-manipulator, 33 degree of freedom robot was designed and built to fly on NASA's space shuttle as a satellite servicing flight demonstration experiment. The robot has sufficient power, speed and reach, to potentially damage or destroy critical components necessary for the shuttle's operation and safe return to Earth. Non-flight versions of this robot have several hundred hours of lab and neutral

buoyancy operational time over the past five years. This provides long term data against which this system safety design process can be evaluated.
The MGA Exoskeleton in the previous case study and the RTSX robot have similar overall system architectures, and both use identical electromechanical actuator and sensor technology. They also use similar control software, with variations only in the specific control modes, device drivers, and the safety checks specific to each robot. Though the hazards listed for each system may appear different, the subtrees contributing to the individual hazards are nearly identical for each robot (e.g. encoders failing and causing uncommanded motion, distributed control components failing to meet their deadline). This strong similarity allows for direct application of the process to both robot systems.
The project safety critiera were driven by a NASA-defined "fail-safe" hazard control approach. This pioneering approach (for NASA) allowed a computer-based control system to have total control of a hazardous payload when traditional approaches are infeasible (Roderick et al., 2004). 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 defined the project safety critiera. The project failure rate criteria were based on the intended 48-hour mission length.
Based on an initial system design, a PHA identified the following three hazards that Ranger presented 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)
Hazard B:	Releasing an untethered object (e.g. an orbital replacement unit) that damages the shuttle or becomes orbital debris
Hazard C:	An object (e.g. an item's restrainting bolt) breaks due to excessive force or torque, and the subsequent pieces damage the shuttle or become orbital debris

Applying FTA to the initial system design, given these three hazards, resulted in the addition of double and triple modular redundancy in certain critical computer components, software algorithms, and sensors. It also resulted in a complete partitioning of system wide safety protocols into an autonomous vehicle-based software safety system that was totally isolated from the rest of the system (Roderick, et al., 2004). This mechanism alleviated the need for safety certification and verification of operators, input devices, communication protocols, and the multitude of operator control stations. This dramatically reduced the complexity of the safety system, which in turn reduced development time, testing effort, and (hopefully) produced a more reliable and safer system.

5.1 Qualitative analysis

The distribution of minimal cut sets by size, for each hazard, is shown graphically in Fig. 14 for the final system design. There are over 3500 minimal cut sets in total, with the smallest minimal cut set size being 2, the maximum size 13, and the average size about 5. As there are no single-component minimal cut sets, this system design satisfies the project's safety criteria. Although there exist double component cut sets, the majority of failure scenarios

leading to a hazard involve three or more events. Fig. 14 also shows that only Hazards A and C have double component cut sets, and Hazard B requires at least three failures before a hazard can occur. There is a significant difference in the number of minimal cut sets for each hazard, largely due to the varying number and size of subtrees for each individual hazard.

5.2 Quantitative analysis

A quantitative probabilistic analysis of RTSX's fault trees was conducted based on limited historical data available from a prototype robot. For this analysis, all RTSX failures were considered non-repairable, since during the mission neither hardware nor software could be repaired nor modified. In computing the hazard probabilities, the assumption of the λ-model that $\lambda t < 0.1$ (in order to simplify $F(t) = 1\text{-}e^{-\lambda t}$) was not valid for the two largest failure probabilities. This assumption was considered acceptable at that time, which has since been validated by operational results.

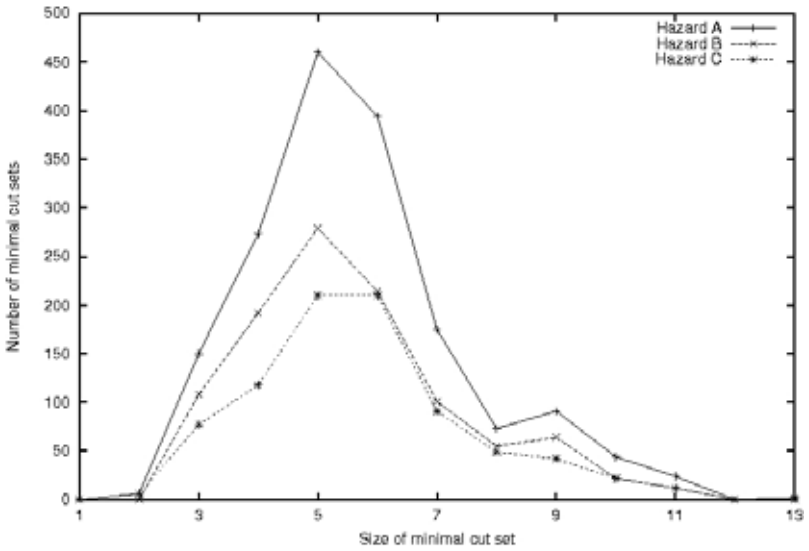

Fig. 14. Distribution of minimal cut set sizes, by hazard.

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. Though this type of probabilistic analysis is known to be conservative, these numbers are uncomfortably high. Examination of the individual subtrees showed that the top five subtrees are significantly more likely to occur then the remaining subtrees. The probabilities of these top five most likely subtrees, and their percentage contribution to their parent hazard's probability, are shown in Table 1. Note that the probability of occurrence in one hour of the remaining subtrees is 0.00005 or lower, or at least two orders of magnitude less likely than any of the top five most likely subtrees. Also note that Hazards B and C are both predominantly caused by one subtree each, i.e., one set of failure combinations. The components and failures making up these individual subtrees could be targetted for significant examination and validation, or potential redesign to reduce their overall probablity of occurence. Note once again that hazard probabilities

calculated from an FTA are only indicative, due to the inherent uncertainty in failure probabilities and the assumptions of the lamba-model.
Sensitivity analysis of the input failure probabilities was conducted on all subtrees, in an effort to identify the effect of their inherent uncertainty. This uncertainty occurs due to assumptions made by manufacturers when defining failure rates for individual commercial off-the-shelf components, as well as from the assumptions and estimates used to extrapolate data from the historical prototype system.

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

Table 1. Estimated probabilities for top five most likely subtrees, and their percentage contribution to the parent hazard's probability.

This analysis individually varied the input failure probability of each failure contributing in a fault tree, and determined the subsequent variation in the parent hazard's probability. Each input failure probability was varied up and down by one order of magnitude. The variation in Hazard C's probability, as a function of the variation in input failure probability is shown in Fig. 15. As expected, the plots indicate a general exponentially increasing effect on the hazard probability as a function of increased individual failure probability. The graph dramatically illustrates that a small number of failure probabilities can significantly increase the overall hazard probability. These individual components could be targeted for additional testing to better determine their predicted failure rates, with a subsequent improvement in the accuracy of the overall hazard probabilities. The graphs for Hazards A and B show the same trends and are not presented here.

5.3 Summary

Despite having fairly high hazard probabilities, RTSX has operated well below these numbers indicating that the computed quantitative probabilities are indeed conservative. In several hundred hours of operation, only one accident has occurred, which was in the form of uncommanded motion. This failure was due to an improperly coded safety check on the bounds of a critical input parameter, and it occurred in the presence of an incomplete and more primitive safety system than was specified in the final system design. The rarity of these events is perhaps an indicator of the overly conservative nature of the probability analysis, as well as an outgrowth of the estimates used in extrapolating failure probabilities from a prototype historical system. The solid operational history may, in fact, validate the system design modifications indicated by this process, and demonstrate the success of the process when applied to a safety critical robot system.

6. Conclusion

A methodology has been presented that can qualitatively and quantitatively evaluate a system design against a set of project safety criteria. This methodology allows system

designers to target individual components or failures, in an effort to make the system more safe. While this methodology cannot produce an absolutely safe system, it provides a mechanism by which a system design can be judged to be sufficiently safe. The methodology also provides for indicative quantitative analysis of a system, which evalutes the system's overall failure rate.

This methodology was applied to two example robotic systems: a shoulder rehabilitation exoskeleton and a multi-arm dexterous space robot. Qualitative analysis of each system allowed for targetted modifications to the system design, producing final systems that were sufficiently safe when judged against the project's safety criteria. Quantitative analysis of the second system indicated uncomfortably high failure probabilities, however, operational data to date indicates the very conservative nature of this analysis, and validates that it can only be used as an indicative evaluation.

Future work for the MGA exoskeleton involves full completion of the FTA, including enumeration of all cut sets, as well as completion of the quantitative analysis. A partial FMEA has already been performed on the MGA Exoskeleton. Completion of this FMEA and comparison to the PHA hazard list may indicate additional hazards, or subtrees of the FTA, that need to be considered. Also, a more comprehensive version of this process would include an FMEA as a useful component. Operational data will continue to be taken for RTSX, and for the exoskeleton once development is complete. Comprehensive comparison of the operational data with respect to the input failure proabilities may help identify certain failures or failure types, that untowardly affect the hazard probability. These variances in the input failure probabilities could be taken into account to improve the accuracy of future quantitative analyses.

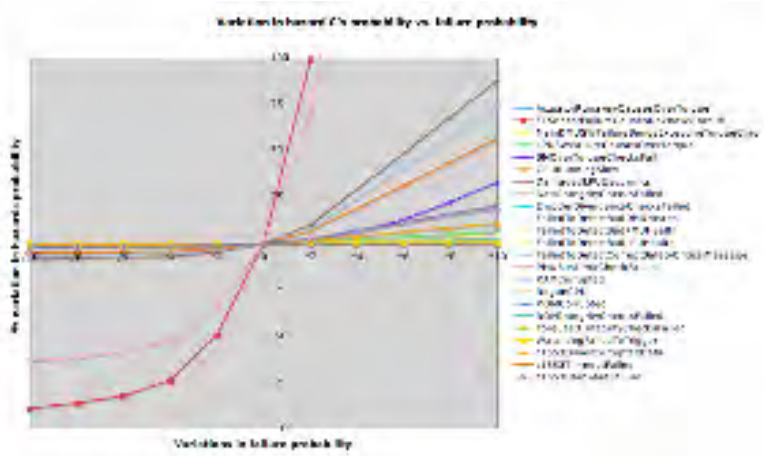

Fig. 15. Percent variation in Hazard C probability as a function of variation in failure probabilities.

7. References

Adler J.R. Jr.; Chang, S.; Murphy, M.; Doty, J.; Geis, P. & Hancock, S. (1997). The Cyberknife: a frameless robotic system for radiosurgery, *Stereotact Funct Neurosurg*, Vol. 69 (1-4 Pt 2), pp. 124-8.

Anderson, T. (1993). Safety-Status and Perspectives, *Proc. of the 12th Int. Conf. on Computer Safety, Reliability and Security*, Janusz Gorski eds., Poznan-Kierkz, Oct. 1993.

Bernhardt, M.; Frey, M.; Colombo, G. & Riener, R. (2005). Hybrid force-position control yields cooperative behaviour of the rehabilitation robot LOKOMAT, *9th Int. Conf. on Rehabilitation Robotics (ICORR)*, Chicago, pp. 536-539.

Blanchard, B. (1991). *Systems Engineering Management*, John Wiley & Sons, New York.

Borras. C. (2006). Overexposure of radiation therapy patients in panama: problem recognition and follow-up measures, *Rev Panam Salud Publica*, Vol. 20, No. 2/3, pp. 173–187.

Bruns, G. & Anderson, S. (1993). Vaildating Safety models with Fault Trees. *Proceedings of 12th International Conference on Computer Safety, Reliability and Security*, Ed. Janusz Gorski, Poznan-Kierkz, Oct. 1993.

Caldwell, D.; Favede, C. & Tsagarakis N (1998). Dextrous exploration of a virtual world for improved prototyping, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 298–303.

Carignan, C.; Liszka, M. & Roderick, S. (2005). Design of an Exoskeleton with Scapula Motion for Shoulder Rehabilitation, *Proc. IEEE Int. Conf. on Advanced Robotics (ICAR)*, Seattle, pp. 524-531, July 2005.

Cavallaro, J. R. & Walker, I. D. (1997). Failure Mode Analysis of a Proposed Manipulator-based Hazardous Material Retrieval System, *Proc. 7th American Nuclear Society Topical Meeting on Robotics and Remote Systems*, Vol. 2, pp. 1096-1102, Augusta, April 1997.

Duchemin, G.; Poignet, P.; Dombre, E. & Pierrot F. (2004). Medically Safe and Sound, *IEEE Robotics & Automation Magazine*, Vol. 11, No. 2, June 2004, pp. 46-55.

Dunn, W. (2003). Designing Safety-Critical Computer Systems, *Computer*, Nov. 2003, pp. 40-46.

Gorski, J.; Maggott, J. & Wardzinski, A. (1995). Modelling Fault Trees as Petri Nets, *Proc. of 14th Int. Conf. on Computer Safety, Reliability and Security*, Oct. 1995.

Guiochet, J. & Vilchis, A. (2002). Safety Analysis of a Medical Robot for Tele-Echography, *2nd IARP IEEE/RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments*, Toulouse, pp. 217-227.

Guthart, G. & Salisbury, J. K. (2000). The intuitive telesurgery system: Overview and application, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 618-622, San Fransisco, Apr. 2000.

He, J.; Koeneman, E.; Huang, H.; Herring, D.; Sugar, T.; Herman, R. & Koeneman, J. (2005). Design of a Robotic Upper Extremity Repetitive Therapy Device, *Proc. Int. Conf. on Rehabilitation Robotics (ICORR)*, Chicago, pp. 95-98.

Hope, S.; Bjordal, E.; Diack, H.; Eddershaw, B.; Joanny, L.; Ortone, G.; Payne, F.; Searson, A.; Sedlacek, K. & Strien, W. (1983). Methodologies for hazard analysis and risk assessment in the petroleum refining and storage industry. *Journal of Hazard Prevention*, Jul/Aug, 1983, pp 24-32.

Knight, J. & Nakano, L. (1997). Software test techniques for system fault-tree analysis, *Proc. of 16th International Conf. on Computer Safety, Reliability and Security*. Ed. Peter Daniel. York, Sept. 1997.

Krebs, H.I.; Volpe, B.T; Ferraro, M.; Fasoli, S.; Palazzolo, J.; Rohrer, B.; Edelstein, L. & Hogan, N. (2002). Robot Aided NeuroRehabilitation: From evidence based to science based rehabilitation, *Topics in Stroke Rehabilitation: Clinical Applications of New Technology*, Vol. 8, No. 4, 2002, pp. 54-70.

Lankenau, A.; Meyer, O. & Krieg-Bruckner, B. (1998). Safety in robotics: the Bremen Autonomous Wheelchair, in *Advanced Motion Control, 1998. AMC '98-Coimbra., 1998 5th International Workshop on*, 29 Jun-1 Jul 1998, pp 524-529.

Leveson, N. (1984). Software Safety in Computer-Controlled Systems, *Computer*, Vol. 17, No. 2, 1984, pp. 48-55

Leveson, N. & Turner, C. (1993). An Investigation of the Therac-25 Accidents, *Computer*, Vol. 26, No. 7, July 1993, pp. 18-41.

Leveson, N. (1995). *Safeware: System Safety and Computers*, Addison-Wesley, Reading, Mass.

Liszka, M. (2006). *Mechanical Design of a Robotic Exoskeleton for Shoulder Rehabilitation*, M.S. Thesis, Dept. of Aerospace Engineering, Univ. of Maryland, Dec. 2006.

Mayr, A.; Mina, S.; Köchln, G.; Kronreif, G. & Saltuari, L. (2006). A New Driven Orthosis for the Upper Extremity (ARMOR): Preliminary Results, *4th World Congress for NeuroRehabilitation, Neurorehabil Neural Repair* 2006, Vol. 20, No. 1.

McDermid, J. (2001). Software safety: Where's the evidence? *Aus Workshop on Industrial Experience with Safety Critical Systems and Software*, Brisbane, July 2001.

Nef, T.; Mihelj, M.; Colombo, G. & Riener R. (2006). Armin – robot for rehabilitation of the upper extremities, *Proc. IEEE Int. Conf. on Robotics and Automation*, Orlando, pp. 3152–3157.

Ozog, H. & Bendixen, L. (1987). Hazards identification and quantification. *Journal of Hazard Prevention*, Sep-Oct, 1987, pp 6-13.

Parnas, D.; Asmis, G. & Madey, J. (1991). Assessment of Safety-Critical Software in Nuclear Power Plants, *Nuclear Safety*, Vol. 32, No. 2, 1991, pp. 189-198.

Potocki de Montalk, J. (1991). Computer Software in Civil Aircraft, *Microprocessors & Microsystems*, Vol. 17, No. 1, 1991, pp. 17-23.

Roderick, S. (2000). *Validation of a Computer-Based Hazard Control System for a Robotic Payload on the Space Shuttle*, M.S. Thesis, Dept. of Aerospace Engineering, University of Maryland, College Park, MD.

Roderick, S.; Roberts, B.; Atkins, E.; Churchill, P. & Akin, D. (2004). An Autonomous Software Safety System for a Dexterous Space Robot, *Journal of Aerospace Computing, Information, and Communication*, AIAA, Dec. 2004.

Roderick, S. & Carignan, C. (2005). An Approach to Designing Software Safety Systems for Rehabilitation Robots, *Proc. of the Int. Conf. on Rehabilitation Robotics (ICORR)*, Chicago, pp. 252-257, June 2005.

Rosen, J.; Perry, J.C.; Manning, N.; Burns, S. & Hannaford, B. (2005). The human arm kinematics and dynamics during daily activities - toward a 7 DOF upper limb powered exoskeleton, *12th Int. Conf. on Advanced Robotics (ICAR)*, Seattle, 2005, pp. 532-539.

Salisbury, J. Kenneth, Jr. (1998). The Heart of Microsurgery, *Mechanical Engineering*, Dec. 1998, p. 46-51.

Shaw, R. (1995). Safety cases – How Did We Get Here? *12th Annual CSR Workshop Safety and Reliability of Software-Based Systems*, Roger Shaw ed., Bruges, Sept. 1995.

Sommerville, I. (1995). *Software Engineering*, 5th ed., Addison-Wesley.

Stephenson, J. (1991). *System Safety 2000*, Von Nostrand Reinhold.

Taylor, R.; Paul, H.; Kazanzides, P.; Mittelstadt, B.; Hanson, W.; Zuhars, J.; Williamson, B.; Musits, B.; Glassman, E. & Bargar, W. (1991). Taming the Bull: Safety in a Precise Surgical Robot, *Proc. of the 5th Int. Conf. on Advanced Robotics: Robots in Unstructured Environments*, Vol. 1, June 1991, pp. 865-870.

Varley, P. (1999). Techniques for Development of Safety-Related Software for Surgical Robots, *IEEE Trans. on Information Technology in Biomedicine*, Vol. 3, No. 4, Dec. 1999, pp. 261-267.

Vesely, W. (1981). *Fault Tree Handbook*, U.S. Nuclear Regulatory Commission.

Voas, J. M. (1995). A Statistial and Automated Code-Based Fault Tree Mitigation Framework for C++, *Safety and Reliabilty of Software-Based Systems, 12th Annual CSR Workshop*, ed. Roger Shaw, Bruges, Sept. 1995.

Weber, W.; Tondok, H. & Bachmayer, M. (2003). Enhancing Software Safety by Fault Trees: Experiences from an Application to Flight Critical SW, *Proc. of the 22nd Int. Conf. on Computer Safety, Reliability and Security*, Edinburgh, Sept. 2003.