Proceedings of the 2005 IEEE
9th International Conference on Rehabilitation Robotics
June 28 - July 1, 2005, Chicago, IL, USA

An Approach to Designing Software Safety Systems for Rehabilitation Robots

Stephen N. Roderick, Craig R. Carignan, *Member, IEEE*

Abstract— **This paper presents a system-level approach to the design of a safety-critical robotic system that is sufficiently safe to satisfy human-subject safety criteria. This system design approach utilizes preliminary hazard analysis, and fault tree analysis, and was successfully applied to a dexterous space robot designed to fly on NASA's space shuttle. An application of this approach to a shoulder rehabilitation exoskeleton will be presented and shown to improve the safety of the overall system.**

I. Introduction

Safety has always been a paramount concern in robotic systems, particularly when there is a potential for humans to enter the robot's work environment. The situation is often avoided by surrounding the workspace of the robot with a detection device that removes power to the robot if someone or something approaches. Unfortunately, this strategy is obsolete when it comes to medical applications such as surgery and rehabilitation where the robot must interact with the patient directly.

In recent years, robots have made substantial in-roads in the medical field. Devices such as Da Vinci [1], CyberKnife [2], and the IM2 Robot [3], have provided researchers and doctors alike with capabilities not previously available. These additional capabilities have also brought with them the issue of individual patient safety. While the robot must enforce the safety of the patient as an object within its workspace, it must also be able to operate upon, or assist the patient, contrary to most robots that are not allowed to affect the reachable human in any manner. This dichotomy creates the need for a safety system that can allow the robot to interact with the patient, but also enforce all necessary safety precautions at the same time.

In perhaps no other application is patient safety more acute than for exoskeletons in which the human is basically encapsulated in a robotic device. The Maryland-Georgetown-Army (MGA) Exoskeleton, shown in Figure 1, is an arm exoskeleton designed to treat shoulder pathology in a collaborative project between Georgetown University and the University of Maryland. The robot has five degrees of freedom, powered by brushless DC motors through a harmonic drive train capable of exerting up to 92 N-m of torque at the shoulder. Encoders mounted on the motors and a suite of force sensors at the shoulder, elbow and wrist provide input to the control system to realize desired rehabilitation protocols.

This basic system does not inherently address the needs of safety, as its design can only identify certain basic robotic failures. The electromechanical subsystems, the software subsystems and the control subsystem, all need to be examined to determine overall patient safety. This paper will detail an approach to generating a sufficiently safe system design for safety-critical rehabilitation applications. The safety system of the MGA Exoskeleton will be used as an illustrative example of this approach.

Previous approaches to safety system design will first be examined in Section II, and an existing approach based on the safety system for a dexterous space robot will be presented in Section III. An overview of the MGA arm and its control system in Section IV will be followed by an example application of this safety system approach in Section V. This section will also examine changes in the initial system design necessary to enforce safety.

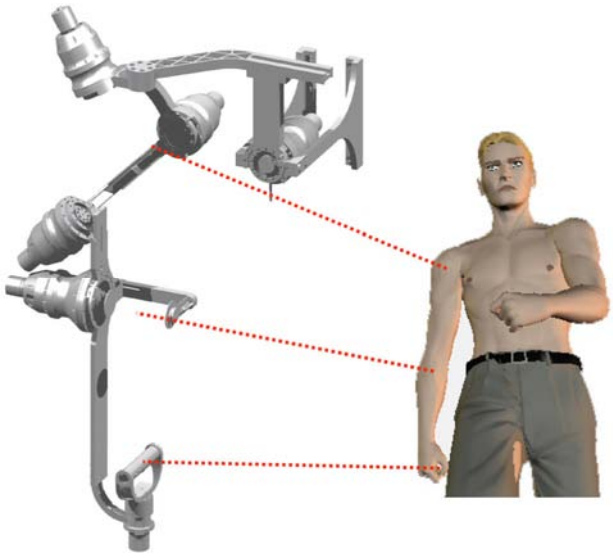

Figure 1 The MGA Exoskeleton designed for shoulder therapy.

II. Previous Work

Previous medical robotics have had to address the issue of patient safety [4] [5]. One of the unique aspects of the medical robotic system presented here, is that the human in the loop is the patient. With surgical or radiological systems

S. N. Roderick is a research engineer with the Space Systems Laboratory, Department of Aerospace Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: snrkiwi-ssl@yahoo.com).

C. R. Carignan, is a research associate with the Imaging Science and Information Systems Center, Department of Radiology, Georgetown University Medical Center, Washington, DC 20057 USA (e-mail: carignan@isis.imac.georgetown.edu).

such as Da Vinci [1] and Cyberknife [2], a patient is being "operated" on by the robot, however, a clinician is directing the robot. With the MGA system, the patient is both the individual upon whom the robot operates, and also the individual who directs the robot.

Unfortunately, given the infancy of this field, there is no industry-standard approach to designing these safety-critical robot systems [4] [5]. Numerous safety-critical software systems have been developed and deployed in other domains ranging from aircraft flight management systems [6] to nuclear power plants [7]. Analytical methods similar to that presented here are a standard and accepted practice in these domains, when identifying and characterizing the likelihood of hazards [8].

This paper presents an approach that was successfully applied to a space robot designed to fly on the NASA space shuttle [9]. This system was the first - and to date only - American robotic system to be certified through three of the four phases of the NASA Space Shuttle Safety Review process. It pioneered a solely computer-based hazard control system for payloads operating on the shuttle.

The following definitions are used in this paper [10]. A "failure" is an abnormal occurrence, while 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.

III. Approach

The process used to determine a sufficiently safe system design is shown in Figure 2. A basic system design that accomplishes the goals of the project is first examined as part of a Preliminary Hazard Analysis (PHA) [11]. A Fault Tree Analysis (FTA) [12] is then developed using the system design and the list of hazards generated by the PHA. The resulting fault trees can be qualitatively examined to determine if the system is "safe enough" for the project's purposes. If not, additional components are typically added to the system in an effort to deal with the specific safety issues raised by the FTA - the system design is modified accordingly - and the process starts again. Once the FTA results show that the project's safety criteria are met, the system design can be considered complete.

The concept of "safe enough" is one that the specific project must establish. 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 safe enough [13] [14]. 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. In order to develop a safe system, it is first important to understand how the system is intended to function.

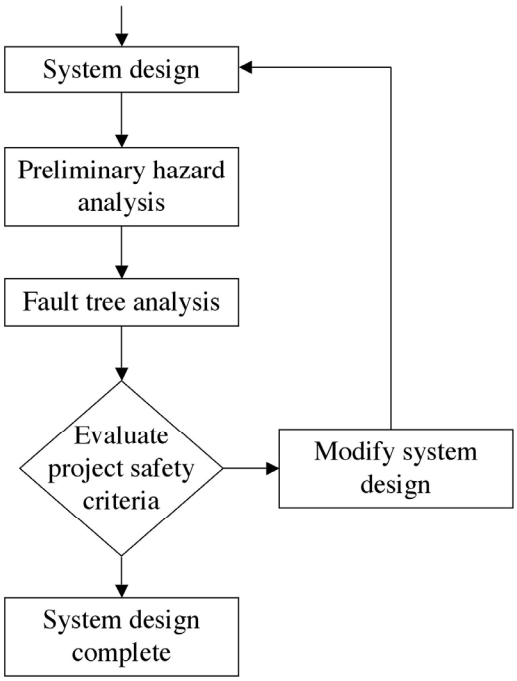

Figure 2 Approach to system design for safety.

IV. Control System

The MGA Exoskeleton has two operating modes: Virtual Reality (VR) Mode, and Physical Therapy (PT) Mode. In VR Mode, the forces exerted at the hand are controlled by interaction with a virtual environment generated by a computer. In PT Mode, the shoulder is exercised about an arbitrary axis through the glenohumeral (GH) joint using a preset resistance profile. In both cases, the scapula joint moves independently to "accommodate" shoulder elevation/depression. The two modes generate the need for contrasting control approaches which are described in more detail below [15].

A. Virtual Reality Mode

Virtual Reality Mode uses computer-generated environments to simulate daily living tasks for functional rehabilitation. The patient views the simulated task and 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's "contact" with the virtual environment and relays them to the controller, which moves the exoskeleton in response to the interaction.

The admittance controller shown in Figure 3 is used to convert sensed contact forces into motion commands [16] [17]. Signals from the gripper force-torque sensor and elbow load cells are input to the virtual environment, which then outputs a desired velocity for the wrist and angular rate of the shoulder-elbow-wrist (SEW) plane roll, ϕ. The desired velocities are then converted into desired joint velocities using the inverse Jacobian, which are then tracked using a proportional-derivative (PD) control law.

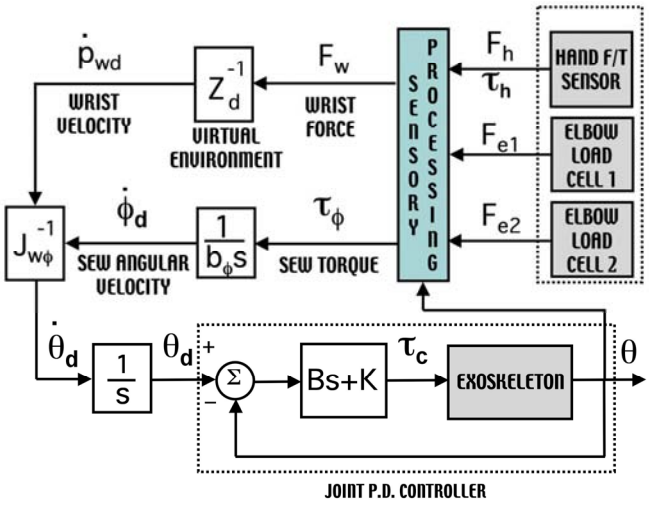

Figure 3 Admittance controller.

B. Physical Therapy Mode

Physical Therapy Mode is basically a programmable resistance trainer that allows the patient to exercise about an arbitrary shoulder rotation axis. For example, to treat rotator cuff injury, therapists often prescribe exercises involving lateral/medial rotation of the shoulder. Since there is no single joint corresponding to shoulder rotation, the exercise involves all three axes of the exoskeleton shoulder. Thus, the controller needs to affect a prescribed resistance profile about the desired axis while preventing rotations about the other shoulder axes.

Since there is no direct way to measure the torques about the shoulder joints, this mode uses an impedance controller with velocity inputs as shown in Figure 4 [18] [19]. In this scheme, the joint velocities are relayed to a Jacobian, J_{GH}, to compute the Cartesian velocities about the GH joint, $_{GH}$. These velocities are then multiplied by the desired resistance profile, which outputs the torques required about the GH axes, τ_{GH}. These torques are then converted into the exoskeleton joint torques using the Jacobian. A dynamic model of the exoskeleton runs in parallel to compute gravity and friction feedforward compensation torques. Since only the shoulder axes are constrained, the elbow pitch is left free to move however the patient desires.

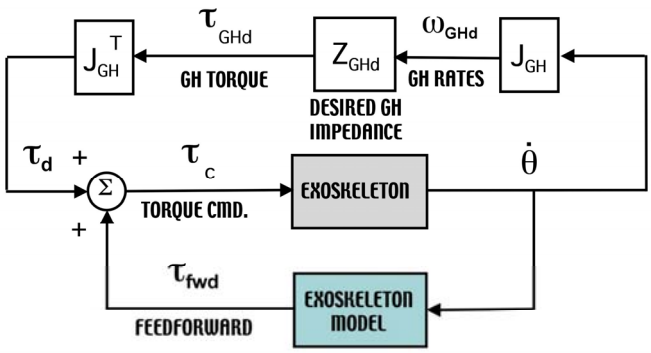

Figure 4 Impedance controller.

In both modes, the scapula joint is controlled independently from the arm joints. The objective is to keep the exoskeleton GH joint as closely aligned with the human GH joint as possible. The desired scapula angle ξ is determined using a biomechanical model of the scapula motion based on the motion of the GH joints. A tracking controller then drives the scapula joint to the desired angle. While only approximate tracking of the GH joint is possible using a single rotary joint, this scenario is preferable to the uncompensated motion observed in simulations of other powered orthoses such as MULOS [20].

The control system defines the minimum suite of sensors and actuators that are required to carry out operations, a partial diagram of which is shown in Figure 5. While this initial system design provides for operational capability, it must be further refined to ensure the patient's safety. The next section will detail an approach to refining a basic system design in order to ensure safety of the patient.

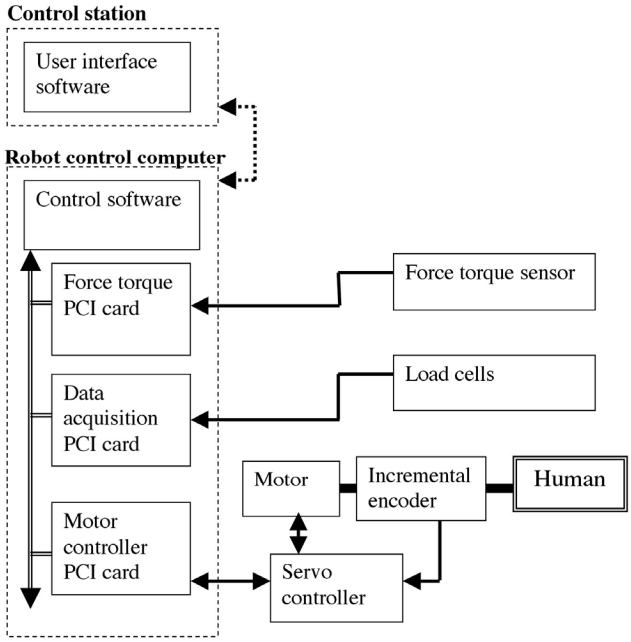

Figure 5 Initial system design. This is the minimum suite of sensors and actuators required for operations. For clarity, only major system components are shown.

V. SYSTEM SAFETY

A preliminary hazard analysis of the MGA system identified three potential hazards (in this context, "excessive" means an unhealthy level, leading to injury):

- Moving the patient outside their safe position range.
- Moving the patient at an excessive velocity.
- Applying excessive torque to the patient or, conversely, allowing the patient to apply excessive torque against the robot.

The project safety criteria specify that no single failure can cause a hazard, and that the system is "fail-safe". A fail-safe system is one that will achieve a safe state in the presence of a detected fault [9]. When a fault is detected, the system will

either a) halt arm motion and hold the current position, or b) safe the arm by removing power to the motors. Removing power has a more pronounced effect on the patient, as they now have to hold up the weight of the device. Thus, this approach is used only when more severe failures occur or when a reliable arm halt cannot be guaranteed. The state of the system, both patient and robotic, is safe for the patient if either a) or b) occurs.

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 will summarily be examined for their causing 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. The symbols used in this work to represent fault tree events and gates are shown in Figure 6. Further details of fault trees and their construction can be found in [12].

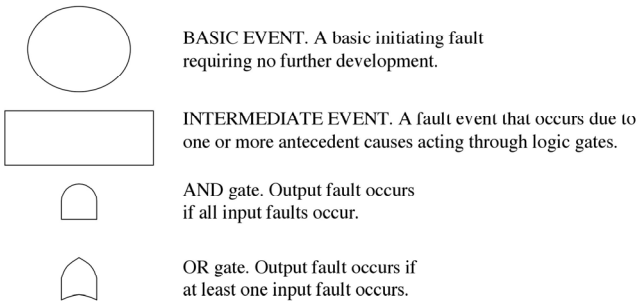

Figure 6 Symbols used in fault trees.

A. Moving the patient outside their safe position range

A fault tree developed from the initial system design of Figure 5, and the top event "Moving the patient outside their safe position range", is shown in Figure 7. This partial fault tree provides an example in which a single fault could cause this hazard. The top event can be caused by any one of numerous possible intermediate events, due to the OR gate attached to the top event. 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 Figure 8, where the shaded components, an absolute encoder and a power amplifier, are additions over the initial system design. Note that for clarity, additional safety components such as emergency stop measures are not shown.

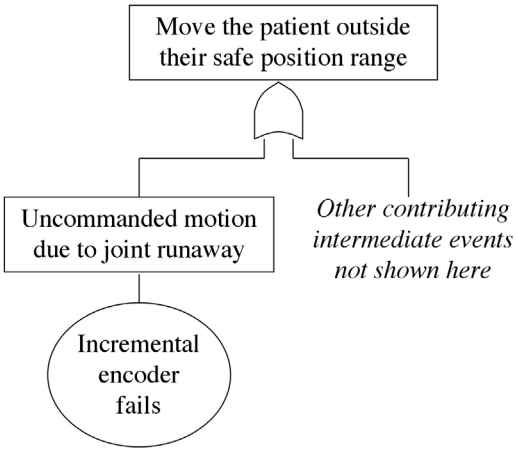

Figure 7 Fault tree for the initial system design and the top event "Moving the patient outside their safe position range". This fault tree shows that a single fault, that of the incremental encoder, could cause the top event to occur.

The fault tree for this top event and the modified system design is shown in Figure 9. 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 fault is capable of producing the hazard.

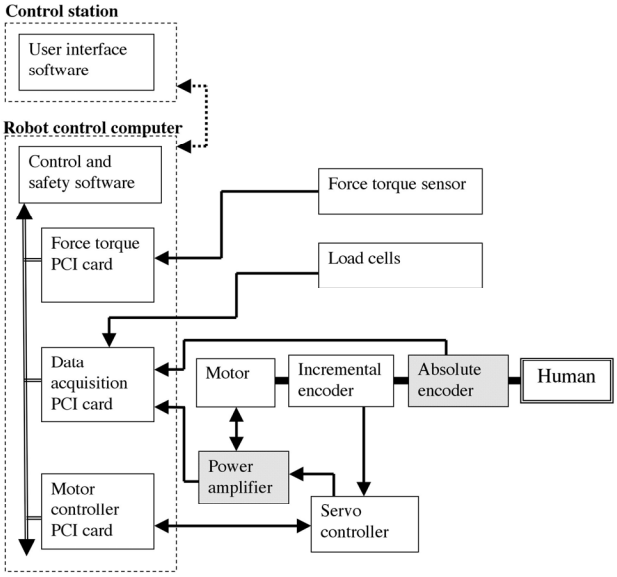

Figure 8 Modified system design with additional components to satisfy project safety criteria. The additions over the initial system design of Figure 5 are shaded. For clarity, only major system components of relevance are shown.

While the modified system design does prevent a single failure from causing this hazard, closer examination of Figure 9 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 paper.

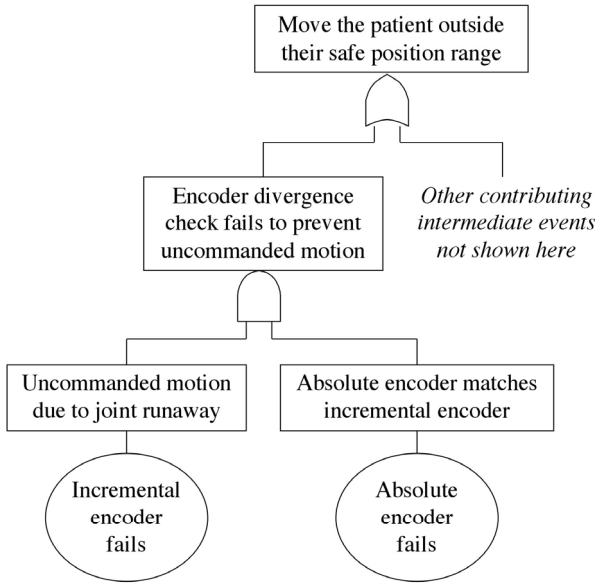

Figure 9 Fault tree for the modified system design and the top event "Moving the patient outside their safe position range". This fault tree indicates that two simultaneous faults are required for the intermediate event shown to cause the top event to occur.

To help determine the overall likelihood of such incredible failures occurring, the fault trees may be quantitatively evaluated. It must be noted, however, that FTA is more a qualitative technique, and that "its absolute accuracy is generally secondary to identification of failure sequences" [21]. Quantitative analysis may therefore be beneficial in simply ranking failures by probabilistic likelihood, versus using the output probabilities as absolute indications of safety [10].

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 further here.

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 Figure 10. A single fault 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 Figure 11 is for the modified system design, and shows the addition of a separate power amplifier with built-in motor current sensor, and a software-based motor power check. 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.

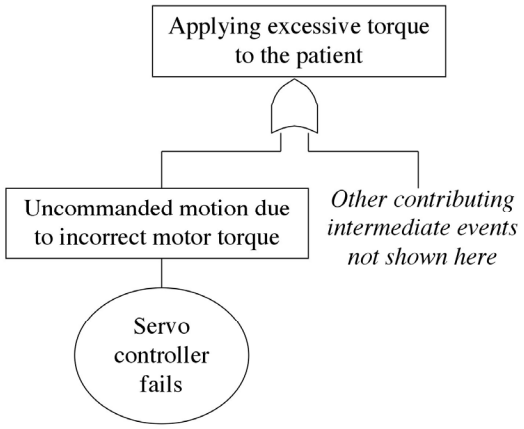

Figure 10 Fault tree for the initial system design and the top event "Applying excessive torque to the patient". This fault tree shows that a single fault, that of the servo controller, could cause the top event to occur.

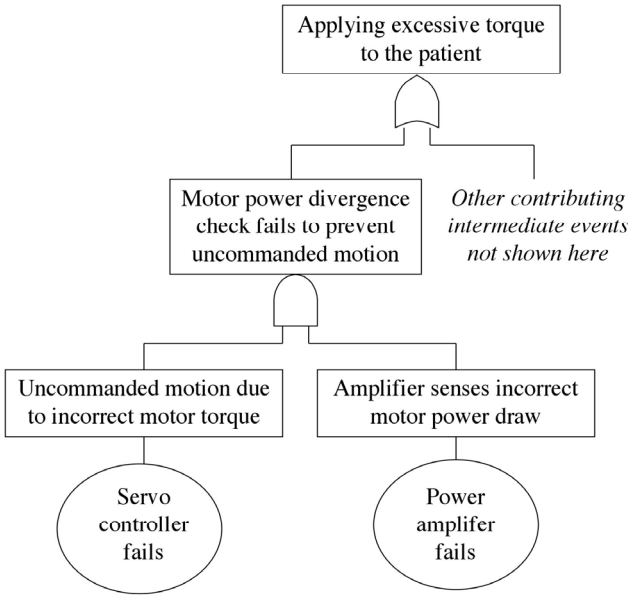

Figure 11 Fault tree for the modified system design and the top event "Applying excessive torque to the patient". This fault tree indicates that two simultaneous faults are required for the intermediate event shown to cause the top event to occur.

VI. Conclusion

The methodology presented here allows system designers to produce an overall system design that is sufficiently safe to satisfy the project's safety criteria. This approach will often result in additional components being added to a system, to ensure that the safety system can detect failures and act accordingly.

The safety system for the MGA Exoskeleton consists of a suite of interwoven hardware constraints and devices (e.g. secondary encoders, slip clutch), electronic checks (e.g. encoder illegal states), and software checks (e.g. encoder divergence checks, electronic component heartbeats). The system design to date satisfies the project safety criteria, and indeed, carries over much of the safety approach and design elements from its space flight predecessor.

Acknowledgment

We would like to acknowledge the rest of the exoskeleton team, Mike, Brian, Walt, John, and JM, who helped make this work possible. This project is being supported by the U.S. Army Medical Research and Materiel Command under Grant #DAMD17-99-1-9022.

References

[1] G. Guthart, J. Kenneth Salisbury Jr, "The intuitive telesurgery system: Overview and application", in Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Fransisco, CA, 2000, pp 618-622.

[2] Accuray, http://www.accuray.com/

[3] Interactive Motion Technologies, http://interactive-motion.com

[4] P. Varley, "Techniques for Development of Safety-Related Software for Surgical Robots", Information Technology in Biomedicine, IEEE Transactions on, Volume 3, Issue 4, Dec. 1999, pp 261-267

[5] R.H.Taylor, H.A. Paul, P. Kazanzides, B.D. Mittelstadt, W. Hanson, j. Zuhars, B. Williamson, B. Musits, E. Glassman, W.L. Bargar, "Taming the Bull: Safety in a Precise Surgical Robot", in *Robots in Unstructured Environments*, Fifth International Conference on Advanced Robotics, volume 1, 19-22 June 1991, pp 865-870.

[6] D. L. Parnas, G. K. Asmis, and J. Madey, "Assessment of Safety-Critical Software in Nuclear Power Plants," *Nuclear Safety,* 32(2), pp. 189-198, 1991.

[7] J. Potocki de Montalk, "Computer Software in Civil Aircraft," *Microprocessors & Microsystems,* 17(1), pp. 17-23, 1993.

[8] W. Weber, H. Tondok, and M. Bachmayer, "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*, Edinburgh, Scotland, 23-26 September, 2003.

[9] S. Roderick, B. Roberts, E. Atkins, P. Churchill, D. Akin, "An Autonomous Software Safety System for a Dexterous Space Robot", Journal of Aerospace Computing, Information, and Communication, AIAA, December 2004.

[10] S. Roderick, "Validation of a Computer-Based Hazard Control System for a Robotic Payload on the Space Shuttle", M.S. Thesis, Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA.

[11] N.G. Leveson, "Safeware: System Safety and Computers", Addison-Wesley, 1995.

[12] W.G. Vesely, "Fault Tree Handbook", US Nuclear Regulatory Commission, 1981.

[13] T. Anderson, "Safety – Status and Perspectives", in *Proceedings of the 12th International Conference on Computer Safety, Reliability, and Security*, Poznan-Kierkz, Poland, 27-29 October 1993.

[14] R. Shaw, "Safety cases – How Did We Get Here?", in *Safety and Reliability of Software-Based Systems*, 12th Annual CSR Workshop, Bruges, 12-15 September 1995.

[15] C. Carignan, K. Cleary: Closed-Loop Force Control for Haptic Simulation of Virtual Environments, Haptics-e, The Electronic Journal of Haptics Research (http://www.haptics-e.org), Vol. 2, No. 2, 1-14, Feb. 2000

[16] J. Maples, and J. Becker, "Experiments in Force Control of Robotic Manipulators", *Proceedings IEEE Intl. Conf. on Robotics and Automation*, April 1986

[17] C. Carignan, D. Akin: Achieving Impedance Objectives in Robot Teleoperation, *Proceedings IEEE Int. Conf. on Robotics and Automation*, Albuquerque, 3487-3492, Apr. 1997

[18] N. Hogan, "Impedance Control: An Approach to Manipulation", Journal of Dynamic Systems, Measurement, and Control, Vol. 108, March 1985

[19] T. Massie and J.K. Salisbury, "The PHANToM Haptic Interface: A Device for Probing Virtual Objects", *Proceedings ASME Winter Annual Meeting: Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Nov. 1994

[20] M. Buckley, and R. Johnson, "Computer Simulation of the Dynamics of a Human Arm and Orthosis Linkage Mechanism", Proc. Instn. Mech. Engrs. Part H, Vo. 211, pp. 349-357, 1997

[21] H. Ozog and L.M. Bendixen "Hazards Identification and Quantification", Hazard Prevention, Sept/Oct 1987, pp 6-13.