Proc. IEEE Annual Reliability and Maintainability Symposium,
pp. 111-116, Philadelphia, PA, January 1997.

Analysis of Robots for Hazardous Environments

Barbara McLaughlin Harpel • University of Virginia • Charlottesville
Joanne Bechta Dugan • University of Virginia • Charlottesville
Ian D. Walker • Rice University • Houston
Joseph R. Cavallaro • Rice University • Houston

Key Words: Coverage, Fault Trees, Robotics

SUMMARY & CONCLUSIONS

Reliability analysis of fault tolerant systems often ignores the small probability that a failure might not be detected or, if detected, may not be properly handled. The probability that the failure *is* detected and properly handled is called *coverage*. Inclusion of coverage in reliability analysis is especially important when analyzing critical systems, systems which for some reason are not easily reparable, or systems whose failure can result in serious damage to the system or its surroundings. One example of a system which can cause such damage is a robot manipulator arm. Robots are being increasingly employed in remote and hazardous environments such as in space and in nuclear waste cleanup, and can exhibit a wild response to subsystem failure, damaging themselves and/or their surroundings. Addition of redundancy to such systems can increase their reliability by allowing continued operation in the presence of faults (provided that the fault is covered), an advantage in a system where repair is difficult or impossible. Coverage models have been used to analyze the behavior of fault-tolerant computer systems in the presence of faults, providing an estimate of the relative probability of an uncovered vs. a covered component failure (given that a fault has occurred) [1]. This paper extends the use of coverage models to the basic components of the joint of a robot and presents data utilizing the calculated coverage for a three-joint robot manipulator arm designed to operate in the plane.

1. *INTRODUCTION*

Fault tolerance has not been an area of major emphasis in the design of industrial robots, and there is not much data available on operational robot failures. However, increasing use of robots for applications in hazardous and/or inaccessible environments such as those in which space exploration, environmental restoration, waste management operations, and some medical applications take place has created a need for safe and reliable robots which can operate over longer periods of time without the need for human intervention.

A robot manipulator arm can be described as a mechanical device consisting of links connected by joints driven by some sort of actuator and whose movement is directed by a controller (a computer-coordinated device) based on the position of the end effector. The number and type of joints required by the arm depend upon the task to be performed and the space in which the robot has to work. For example, a robot designed to operate in a plane requires a minimum of two joints in order to cover that space but may need more if it must also negotiate around some sort of obstacle, while a robot designed to operate in a three-dimensional space requires a minimum of three joints. Figure 1 shows two types of joint while Figure 2 shows a manipulator arm, designed to operate in the plane, which has more than the required two joints so that it can successfully maneuver around the obstacle in the center. A robot which possesses more than the required number of joints to operate in its task space is termed *kinematically redundant.*

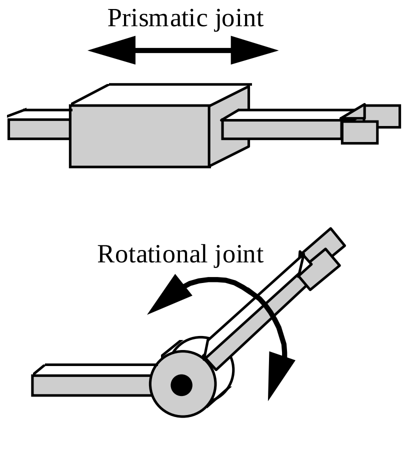

Figure 1. Types of joint.

The basic components of a robot joint are a motor to provide motion and some sort of sensor whose output is used by the robot's controller to determine position and speed both at the joint and at the end of the manipulator arm itself. Sensors can be of several types: position sensors (either translational or rotational), force/torque sensors, speed sensors, or acceleration sensors [3]. Figure 3 shows the fault tree for a basic joint.

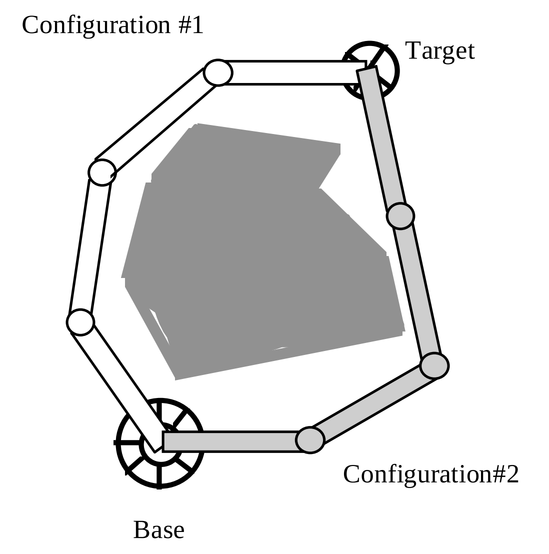

Figure 2. Kinematically redundant planar robot.

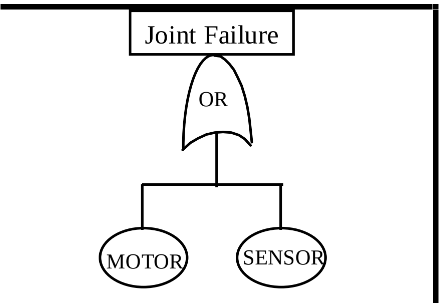

Figure 3. Fault tree for a basic joint.

Physical redundancy can be implemented in a robot manipulator arm in two ways. A component of the joint can be replicated, so that, for instance, there are two sensors or motors in parallel where before there was only one sensor or motor, or an entire extra joint can be included (kinematic redundancy, defined above). If a robot has more than the minimum number of joints it can choose between several configurations of the joints to position the end effector at a particular location (refer again to Figure 2). When a robot is designed for operation in the plane, inclusion of a third joint also allows the robot to continue to perform its function even after failure of one joint; thus kinematic redundancy, when managed properly, also provides a measure of fault tolerance. (Kinematics is the mathematical process relating the end effector position and orientation to the joint configuration [3].) One of the dangers of joint failure is erratic swinging behavior of the free joint if it is not locked in position, possibly causing serious damage to objects in the immediate vicinity including the robot itself if not checked quickly [4]. The ability of the joint to be locked in place is extremely important to the reliability and safety of the system.

A word about robotic control is necessary before we proceed with the description of the system analyzed. The robot's controller directs the robot's movement through its workspace based on a plan of the desired position or trajectory of the robot end effector. An inverse kinematics algorithm is used to compute the next desired joint configuration given the current configuration and the next desired end effector position. The model used to calculate this position can also be used to detect faults. The difference between the position predicted and the actual position (if outside of some threshold value -- exact agreement is usually impossible due to sensor and modeling errors inherent in the fault-free system which arise, for example, from linearization of the robot equations and inaccuracies in model parameters such as link inertia or mass [3]) can indicate that an error has occurred.

2. *SYSTEM ANALYZED*

The system analyzed in this paper is a three-joint kinematically redundant robot manipulator arm designed to operate in the plane. As can be observed from the fault tree for the system shown in Figure 4, only two working joints are required for continued operation. Each joint has two sensors and one motor. A similar robot was shown in [5] (the controller was not included in the analysis; otherwise the system studied in [5] is identical to that studied here) to exhibit much higher reliability than a basic two-joint robot with one sensor and one motor at each joint.

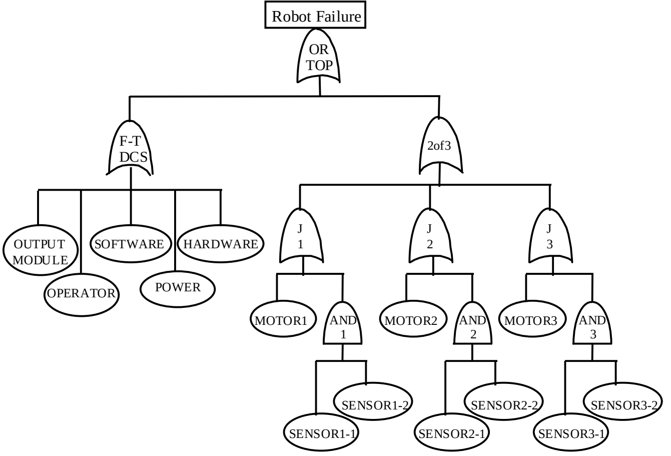

Figure 4. Fault tree for three-joint kinematically-redundant robot designed to operate in the plane.

In order to make the system analysis more complete, we have included here a fault-tolerant digital control system necessary for the operation of the robot. Failure rates and coverage information for the controller are extrapolated from [2], one of the few studies available which uses actual failure data as its source. Our figures come from the data on dual-redundant processor/triple-redundant input module logic (DUAL/TRIML) controllers. These data are from nuclear power plants and process control applications, not robotics, but they provide a good starting point for analysis. The study also includes operator error and power supply failure as causes for system failure; since many robotic applications also require input from a human operator, these data were deemed appropriate for demonstration purposes. As in [2] we have separated the failure modes of the FT-DCS into five categories: output module failure (basic event output), failures caused by operator error (basic event operator), failures caused by software error (basic component software), power supply failures, and hardware failures (basic event hardware). Failure rates used for the basic components of the robot joints (calculated from probability data in [5], using base failure rate) and for the controller (from [2]) are shown in Table 1.

Gate	Basic Component	Failure Rate
Joint	motor	9.24E-6
	sensor	1.55E-5
Controller	output	1.0975E-5
	operator	9.22E-6
	software	9.22E-6
	power	4.829E-6
	hardware	9.658E-6

Table 1. Failure rates for basic components of robot fault tree.

3. *COVERAGE MODELS*

A fault in some system component does not necessarily lead to the component's failure since the monitoring computer system may recognize and handle a transient fault and allow the component to continue operating as if the fault had never occurred. Thus there is more than one failure mode exhibited by the component; these failure modes and methods for handling the problem must be determined.

Dugan and Doyle [1] describe the general model of a recovery process begun when a fault occurs as shown in Figure 5. Occurrence of a fault provides entry into the model while the three exits signify three possible outcomes. The transient restoration exit (labeled *R*) represents the correct recognition and recovery from a transient fault. A transient is usually caused by external or environmental factors, such as a power line glitch or excessive heat. The general consensus is that most faults are transient. In the case of a transient fault, an instruction may be re-tried or a new sensor reading may be taken, etc.; successful recovery from a transient fault restores the system to a consistent state without discarding any components. To reach this exit, the system must both detect the error produced by the fault in a timely fashion and perform the correct recovery procedure, and the fault must disappear quickly.

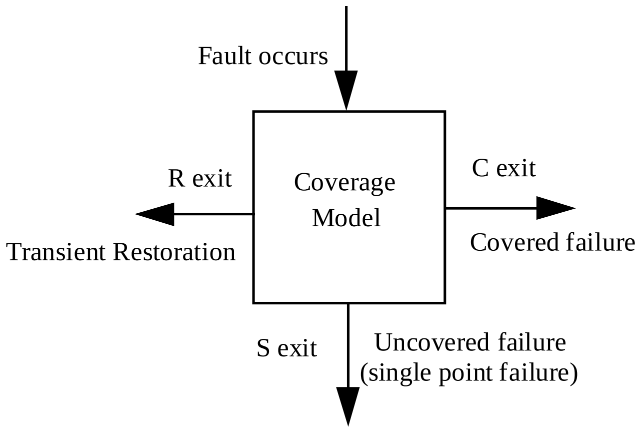

Figure 5. General structure of a coverage model [1].

To reach the exit labeled *C* (the permanent coverage exit), the fault must be determined to be permanent and the faulty component must be successfully isolated and removed from the system. If an undetected error propagates through the system, or if the faulty component cannot be isolated, the system cannot be reconfigured and the single point failure exit (labeled *S*) has been reached; a single fault will cause the system to crash. The distinction should be made between a fault and a failure in a single component. If a fault affects a component, the component does not necessarily fail because the fault may be transient. A component failure occurs only if the transient restoration is unsuccessful and thus the transient restoration exit is not reached. If the permanent coverage exit is reached instead, then a covered component failure has occurred. A covered component failure does not necessarily cause system failure; system failure is dependent upon the remaining redundancy in the system. If the single point failure exit is reached, then an uncovered component failure has occurred.

Construction of a coverage model involves several steps: 1.) determining the possible failure modes for the component being analyzed; 2.) assigning relative probabilities to those failure modes and determining how likely their detection; 3.) defining the proper method for handling the failure (if it is detected); and 4.) calculating or estimating the likelihood that the correct steps are taken to recover from the failure [1].

As described in the previous section, our robot joint consists of two types of component: motor (one) and sensor (two). A coverage model is constructed for each type of component. First we shall consider the sensor.

Common failure modes for a joint sensor include frozen (sensor produces a constant value), biased (true value is different by some constant amount from the value the sensor reports), run-away (erratic), and spike (sudden high or low) [3], [4]. In all cases the first question asked is whether the failure is detected and with what probability. If the relative proportions of the failure modes are unknown, equal probabilities might be assumed as a first estimate and then updated as data becomes available, or experience with some similar system may be used. Any failure which goes undetected is an uncovered failure and is assumed to lead to system failure [1]. For any failure mode an appropriate action to be taken if the failure is detected must also be determined. If this action is not taken for any reason, the failure is considered *uncovered*, but if the proper action is taken successfully the failure is *covered*. For a sensor, handling the frozen and run-away modes requires ignoring the data in subsequent calculations, in other words considering the sensor failed and ignoring its input to the controller. If the biased failure mode is detected, simply removing the bias allows continued use of the sensor and so there is *transient restoration*. A transient restoration is also reached in the case of a spike; if on subsequent iterations of the control algorithm the sensor reports a value which agrees with that predicted by the model the sensor is considered still in working order, and the transient restoration exit of the model has been reached.

The coverage model for the joint sensor is illustrated in Figure 6. The top level represents occurrence of some fault resulting in the sensor entering one of the failure modes. The decision point following the failure modes asks with what probability the failure is detected. If the failure is not detected then the fault is uncovered and the single point failure exit is reached. If the fault is detected, the next decision point asks whether or not the proper action is taken. If the proper action is not taken, the failure is uncovered and again the single point failure exit is taken. If the proper action is taken, either the fault is covered or transient restoration has occurred.

Probabilities for each path are shown at the branch points. Coverage can be calculated by determining the probability of reaching each endpoint, then adding together like failure-type probabilities. For instance, the probability that a failure occurs in the frozen mode *and* is properly handled is the probability that the failure is detected and properly handled or (0.6)(0.99)(0.99) = 0.58806; this is a covered failure.

The probability of an uncovered failure in the frozen mode is the probability that the failure is not detected plus the probability that if detected, the failure was not handled properly, or (0.01)(0.6) + (0.01)(0.99)(0.6) = 0.01194.

The coverage factor for the joint sensor is the probability that the fault is detected and handled properly. It can be calculated by summing the end probabilities of all those paths which result in that outcome (covered failure and transient restoration), or alternatively by subtracting the sum of the probabilities of an uncovered failure from 1. The coverage for the joint sensor under the assumptions and conditions stated is thus calculated as 0.96054.

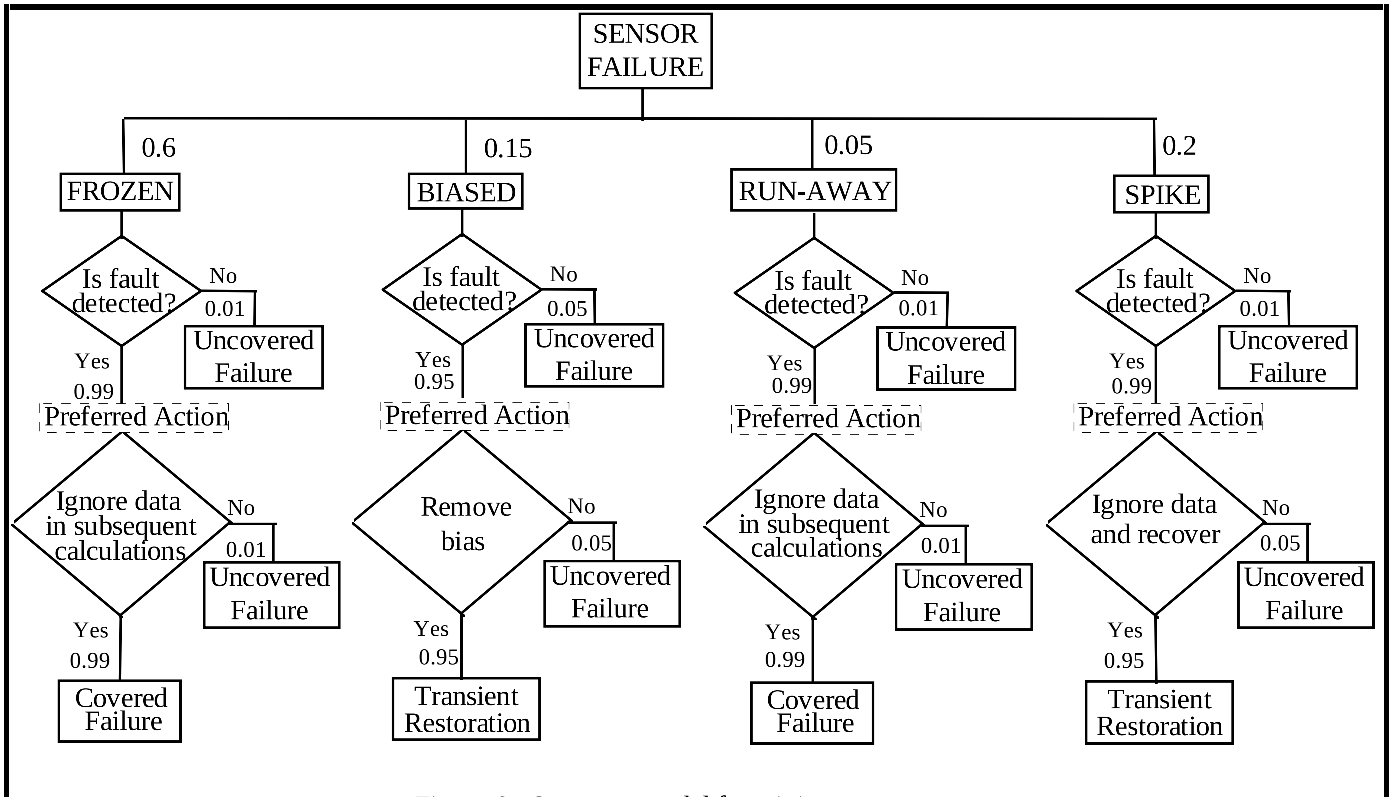

Figure 6. Coverage model for a joint sensor.

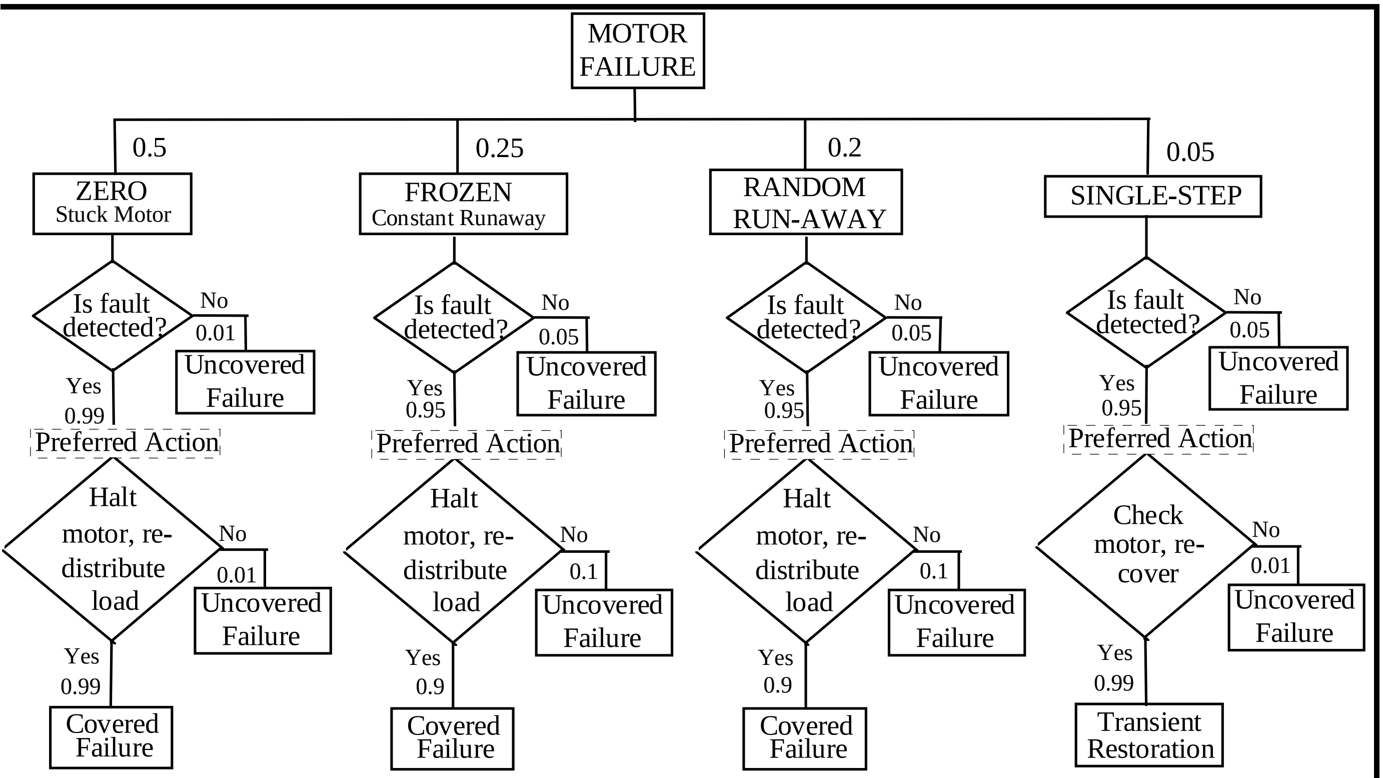

Figure 7. Coverage model for a joint motor.

The failure modes for a joint motor include zero, frozen, random run-away, and step failure. The zero mode represents a stuck motor -- it's not able to move at all. The frozen mode is also termed constant run-away -- it has only one response: run at a constant pace. A motor in the random run-away mode, on the other hand, will exhibit erratic response, or multiple step-type errors in the torque of the motor. For a joint with just one motor, these first three failure modes are handled identically: halt the motor (locking the joint in position) and redistribute the load, in other words allow the remaining joints to handle movement. (There has been some work done using two motors per joint; in this case, if one motor is in the constant runaway failure mode the other can be operated to counteract the failed motor and so there is a transient restoration [3].) In the single-step failure mode (a sudden increase or decrease in the torque provided), however, recovery is possible if the fault is detected, so for the system analyzed here the transient restoration exit can be reached for this failure mode only.

Figure 7 shows the coverage model for a joint motor, developed in a similar manner as for the sensor with the exception that with only one motor there is transient recovery only for the step failure mode. The varying probabilities of fault detection reflect the fact that it is easier to detect a torqueless joint than one with active torques. Likewise, the higher probability of not covering a failure in the frozen and run-away modes reflects the difficulty in stopping a run-away or constantly moving joint -- brake failure and related problems are more likely. The coverage for the motor is calculated to be 0.921825.

4. *RESULTS*

Once the coverage factors to be included were determined the fault tree was solved. For comparison purposes, several permutations of the robotic system were also solved: the basic robot, just two joints, with and without the controller and with and without coverage; the two-jointed robot with redundant sensors (2 sensors, 1 motor at each joint, only two joints), with and without the controller and with and without coverage; and the system as described in Figure 4, both including and excluding the controller and including and excluding coverage. (Note that *not* including coverage is identical to assuming that coverage is perfect.) The results are summarized in Table 2. It should be noted that the inclusion of the controller vastly decreases the effect of including coverage on the joint components; this is because the failure rate for the controller is dominating. The inclusion of coverage for the joint components when the controller is excluded from the system causes the unreliability to increase by a factor of approximately 22.5. If the failure data for the fault tolerant digital controllers considered here is comparable to those used in robotic controllers, these results indicate that as much attention should be paid to the reliability (or lack thereof) of the controller as to the reliability of the joint components.

UUNRELIABILITY		Two-joint, no redundancy	Two-joint, redundant sensors	Three-joint, redundant sensors
Controller	No Coverage	0.0891546	0.0609206	0.0432061
	Coverage	0.0891545	0.0631644	0.0487023
No Controller	No Coverage	0.048257	0.0187747	0.0002652
	Coverage	N/A	0.0211193	0.0060081

Table 2. Unreliability of various robot configurations.

ACKNOWLEDGMENT

The models were solved using DIFtree, a software package developed at the University of Virginia under the sponsorship of NASA Langley Research Center.

REFERENCES

1. J. Bechta Dugan and S.A. Doyle. New Results in Fault-Tree Analysis. Tutorial presented to 1996 Reliability and Maintainability Symposium, January 1996.

2. H.M. Paula, M.W. Roberts, and R.E. Battle. Operational Failure Experience of Fault-Tolerant Digital Control Systems. In *Reliability Engineering and System Safety* **39**: 273-289, 1993.

3. M.L. Visinsky, J.R. Cavallaro and I.D. Walker. Robotic Fault Detection and Fault Tolerance: A Survey. In *Reliability Engineering and System Safety* **46**(2): 139-158, 1994.

4. M. L. Visinsky, J. R. Cavallaro and I. D. Walker. A Dynamic Fault Tolerance Framework for Remote Robots. In *IEEE Transactions on Robotics and Automation*, **11**(4): 477-490, 1995.

5. I. D. Walker and J. R. Cavallaro. The Use of Fault Trees for the Design of Robots for Hazardous Environments. In *Proceedings of the IEEE Reliability and Maintainability Symposium*, pages 229-235, January 1996.

BIOGRAPHIES

Barbara McLaughlin Harpel, *ME*
2824 Glen Gary Drive
Richmond, Virginia 23233 USA
Internet (e-mail): bharpel@richmond.infi.net, bmh2c@virginia.edu

Barbara McLaughlin Harpel was awarded the BS degree in Biology from the Pennsylvania State University, University Park, PA, in 1975, and has just completed a Master's of Engineering degree in electrical Engineering at the University of Virginia. She is currently spending time at home with her two daughters but plans to return to the workplace in the spring. Her professional interests include fault tolerant computing, hardware and software reliability engineering, and the tools used for reliability analysis. Ms. Harpel is a member of Eta Kappa Nu and Tau Beta Pi.

Joanne Bechta Dugan, *PhD*
Department of Electrical Engineering,
University of Virginia
Charlottesville, Virginia 22903-2442
Internet (e-mail): jbd@virginia.edu

Joanne Bechta Dugan was awarded the BA degree in Mathematics and Computer Science from La Salle University, Philadelphia, PA in 1980, and the MS and PhD degrees in Electrical Engineering from Duke University, Durham, NC in 1982 and 1984, respectively. Dr. Dugan is currently Associate Professor of Electrical Engineering at the University of Virginia, and was previously Associate Professor of Computer Science at Duke University and Visiting Scientist at the Research Triangle Institute. She has performed and directed research on the development and application of techniques for the analysis of computer systems which are designed to tolerate hardware and software faults. Her research interests thus include hardware and software reliability engineering, fault tolerant computing, and mathematical modeling using dynamic fault trees, Markov models, Petri nets and simulation. Dr. Dugan is an Associate Editor of the IEEE Transactions on Reliability, is a senior member of the IEEE, and is a member of Eta Kappa Nu and Phi Beta Kappa. She is serving on the National Research Council Committee on Application of Digital Instrumentation and Control Systems to Nuclear Power Operations and Safety.

Ian D. Walker, *PhD*
Department of Electrical and Computer Engineering
Rice University
Houston, Texas 77251-1892
Internet (e-mail): ianw@rice.edu

Ian D. Walker received the B.Sc. degree in Mathematics from the University of Hull, England, in 1983. He received the MS degree in 1985, and the PhD in 1989, both in Electrical Engineering, from the University of Texas at Austin. In 1989 he joined the faculty of Rice University, Houston, TX, where he is an Associate Professor of Electrical and Computer Engineering. His research interests are in the areas of robotics and control, particularly fault tolerant robot systems; robotic hands and grasping; and kinematically redundant robots.

Joseph R. Cavallaro, *PhD*
Department of Electrical and Computer Engineering
Rice University
Houston, Texas 77251-1892
Internet (e-mail): cavallar@rice.edu

Joseph Cavallaro received the BS degree from the University of Pennsylvania, Philadelphia, PA in 1981, the MS degree from Princeton University, Princeton, NJ, in 1982, and the PhD degree from Cornell University, Ithaca, NY, in 1988, all in electrical engineering. From 1981 to 1983, he was with AT&T Bell Laboratories, Holmdel, NJ. In 1988 he joined the faculty of Rice University, Houston, TX, where he is an Associate Professor of Electrical and Computer Engineering. His research interests include computer arithmetic, fault tolerance, VLSI design and microlithography, and VLSI architectures and algorithms for parallel processing and robotics.