# Classifying and Recovering from Sensing Failures in Autonomous Mobile Robots

**Robin R. Murphy** **David Hershberger**
Center for Robotics and Intelligent Systems
Colorado School of Mines
Golden, CO 80401-1887
phone: (303) 273-3874 fax: (303) 273-3975
{rmurphy,dhershbe}@mines.edu

### Abstract

This paper presents a characterization of sensing failures in autonomous mobile robots, a methodology for classification and recovery, and a demonstration of this approach on a mobile robot performing landmark navigation. A sensing failure is any event leading to defective perception, including sensor malfunctions, software errors, environmental changes, and errant expectations. The approach demonstrated in this paper exploits the ability of the robot to interact with its environment to acquire additional information for classification (i.e., active perception). A Generate and Test strategy is used to generate hypotheses to explain the symptom resulting from the sensing failure. The recovery scheme replaces the affected sensing processes with an alternative logical sensor. The approach is implemented as the Sensor Fusion Effects Exception Handling (SFX-EH) architecture. The advantages of SFX-EH are that it requires only a partial causal model of sensing failure, the control scheme strives for a fast response, tests are constructed so as to prevent confounding from collaborating sensors which have also failed, and the logical sensor organization allows SFX-EH to be interfaced with the behavioral level of existing robot architectures.

## Introduction

The transfer of autonomous mobile robot (AMR) technology to applications in manufacturing, defense, space, hazardous waste cleanup, and search and rescue missions has been impeded by a lack of mechanisms to ensure robust and certain sensing. The actions of an AMR depend on its perception; if perception is faulty and goes unnoticed, the robot may "hallucinate" and act incorrectly. One key mechanism for robust sensing is fault-tolerance: the ability to detect sensing failures and either recover from them in such a way as to allow the robot to resume performance of its task(s) or to gracefully degrade.

Previous work in robotic sensing has demonstrated how certain types of sensing failures can be detected either at the behavioral (i.e., self-monitoring) (Ferrell 1993; Murphy & Arkin 1992) and/or deliberative layer (i.e., global monitoring) (Hughes 1993; Noreils & Chatila 1995). An open research question is how to recover from these failures. In the general case, recovery requires identification of the source of the problem; *if the cause is not known, the wrong response may be employed.* Detection of a failure does not necessarily mean that the cause is known. For example, in (Murphy 1992), three different problems which interfered with sensing in a security robot (sensor drift, incorrect placement of the robot, sensor malfunction) evinced that same symptom: a lack of consensus between the observations. The appropriate response to each problem was significantly different (recalibrate the offending sensor, rotate the robot until it reached the correct view, and replace the damaged sensor with an alternative, respectively). However, the correct response was known once the cause was identified. While classification is essential for the general case, it may be unnecessary in situations where the recovery options are limited, i.e. "do whatever works" (Payton *et al.* 1992).

This paper presents a symbolic AI approach to classifying and recovering from sensing failures. The characteristics of the AMR domain is differentiated from typical diagnosis applications (e.g., medicine, geological interpretation) in the next section. Related work in problem solving and diagnosis for robotic sensing follows. An overview of the approach taken in this paper is given next. Classification of errors is done with a novel extension of the basic Generate and Test strategy developed for Dendral (Lindsay *et al.* 1980), with contributions from Generate, Test, Debug (Simmons & Davis 1987). This classification scheme takes advantage of the robot's ability to actively use other sensors and feature extraction algorithms to test hypotheses about the sensing failure; it can be considered a form of active perception (Bajcsy 1988). The classification and recovery scheme is implemented as the exception handling (EH) portion of the Sensor Fusion Effects (SFX) architecture. Demonstrations of SFX-EH on a mobile robot with a landmark navigation behavior are reviewed. The paper concludes with a summary and brief discussion, including on-going research efforts.

## Sensing Failures in AMR

A characterization of sensing failures in AMRs is useful at this point for two reasons. First, it provides the context for justifying the approach taken in this paper. Second, it will distinguish classifying and recovering from sensing failures for AMRs from the connotations associated with general diagnosis in other domains such as medicine and the identification of geological features. The unique attributes of this domain are:

*The class of sensing failures includes more than sensor failures.* For the purposes of this paper, a *sensing failure* is defined as any event leading to defective perception. These events may stem from sensor hardware malfunctions, bugs in the perceptual processing software (e.g., does not work in a particular situation), changes in the environment which negatively impact sensing either at the hardware or software level (e.g., turning the lights off), or errant expectations (e.g., looking for the wrong thing at the wrong time).

The inclusion of software defects, environmental change, and errant expectations as sources of sensing failures makes classification particularly challenging. Indeed, one of the motivations for (Payton *et al.* 1992) is to avoid having to attempt to identify software defects. These sources of faulty sensing have the potential to interrupt progress, especially changes in the environment. Exploiting the environment is a fundamental principle of behavioral robotics. However, (Howe & Cohen 1990) note the difficulty of designing agents that can tolerate environmental change. Since AMRs function in an open world, this suggests that this difficulty will be exacerbated and environmental change will be a significant source of problems as robots are deployed in more demanding settings.

*Sensing failures occur frequently, but different types occur infrequently.* (Ferrell 1993) noted that in experiments with Hannibal, a hexapod robot with over 100 sensors, a hardware sensor failure occurred approximately once every two weeks. Our experience with two different mobile robots is consistent.

*It is unrealistic and undesirable to attempt to explicitly model all possible failure modes.* (Velde & Carignan 1984) devised one such explicit modeling scheme. However, this scheme assumed that all sensors were of the same type and their observations could be correlated statistically. But it begs the issue of how to acquire statistical data about a set of events, when, by definition, the very members of that set may not be known *a priori*. The difficulties are increased as roboticists turn to multiple sensors (sensor fusion). Modeling the interactions between sensors for the environment and the task leads to a combinatorial explosion with a statistical method such as (Velde & Carignan 1984; Weller, Groen, & Hertzberger 1989), again ignoring that a sensing failure may result from a never encountered or unanticipated event. Even if multi-sensor modeling could be done satisfactorily, the causal models are unlikely to be portable to new sensor configurations and application domains.

*An AMR can actively perceive.* One advantage that an AMR has is that it can acquire new information by deliberately engaging its environment as per *active perception* (Bajcsy 1988), and/or by extracting new meanings from previous observations (e.g., examines the recent history of measurements).

*An AMR may have both redundant and complementary sensing modalities.* The trend in robotic sensing is to use a small set of general purpose sensors. Some sensors may be redundant (i.e., two or more of the same sensor). However, the majority of sensors are likely to be complementary. For example, at the AAAI Mobile Robot Competitions, the entries are invariably equipped with vision and sonar. This makes classification challenging because the scheme cannot assume that there is an alternative sensor which can directly collaborate a suspect sensor; instead, inferences from the behaviors of other sensors will have to be made.

*Exception handling is a secondary function in an AMR.* In other domains, diagnosis is their primary task. In an AMR, sensing failures can be viewed as *exceptions* which cause the robot's progress to be suspended. Reliable sensing must be reestablished before the robot can resume the behavior and complete the intended task. However, an AMR may have only a finite time to spend on exception handling. It can't remain indefinitely in a hostile environment such as Three Mile Island or an outgassing Near Earth Object without increasing the risk of a hardware failure from radiation or catastrophe. Therefore, the time dedicated to exception handling is an important consideration in the development of any classification and recovery scheme.

*Exception handling must be integrated with the whole system.* To see how sensing failures impact the whole system, consider the following examples. First, because the robot cannot act correctly without adequate sensing, an AMR must cease execution of the failed behavior and possibly revert to a stand-by, defensive mode if it cannot continue other behaviors. This requires information about sensing failures to be propagated to the behavioral or task manager. If the behavior cannot recover quickly, the mission planner aspect of the robot must be informed so that it can replan or abort the mission. Second, since classification and recovery may involve active perception, contention for sensing resources may occur, e.g., is it safe to take away sensor X from behavior Y and point it in a different direction? Contention resolution requires knowledge about the robot's goals, interchangeability of sensors, etc., making exception handling a process which must communicate with other modules in the robot architecture. Third, if the source of a sensing failure is used by other behaviors, the recovery scheme should include replacing the failed component in the other behaviors which may be hallucinating, as well as the behavior that first detected the problem.

The attributes of the classification and recovery task for AMR itemized above lead to a characterization of a desirable exception handling mechanism and an appropriate problem solving strategy. This exception handler is intended to be applicable to any AMR sensing configuration. Because sensor failures occur frequently and suspend progress of the robot, exception handling must attempt to effect a timely recovery. The exception handler must interact with the task manager to prevent unsafe actions from occurring during classification and recovery. The exception handling scheme can reduce the down time by exploiting any situations where a recovery scheme can be directly invoked, either because the symptom clearly defines the cause or because all possible causes result in the same response. It should continue to attempt to identify the source of the sensing failure in a background mode if it invokes a direct recovery scheme. The exception handler can use active perception to overcome the open world assumption and the resultant difficulty in constructing a complete model of failure modes. But active perception leads to a new issue of how to safely reallocate sensing resources from other behaviors (if needed) to identify the source of the problem. Therefore, the exception handling mechanism must be a global, or deliberative, process in order to reason about possible corroborating sensors which may not be allocated to it. These sensors may or may not be redundant. The mechanism must be able to handle contention resolution or communicate its needs to the appropriate sensor allocation process. When the exception handler identifies the source of the failure, it propagates the information to other behaviors so they don't hallucinate or go into a redundant classification and recovery cycle.

## Related Work

As noted in the introduction, detection, classification, and recovery from sensing failures in mobile robots has been addressed by (Noreils & Chatila 1995), (Ferrell 1993) and (Payton *et al.* 1992). Other noteworthy efforts are those by (Weller, Groen, & Hertzberger 1989), (Velde & Carignan 1984), (Hanks & Firby 1990), and (Chavez & Murphy 1993).

(Weller, Groen, & Hertzberger 1989) and (Velde & Carignan 1984) deal with sensor errors in general. (Weller, Groen, & Hertzberger 1989) create modules for each sensor containing tests to verify the input based on local expert knowledge. Environmental conditions determine whether a test can be performed or not. The partitioning of problem space by symptom is based on these modules. The approach taken in this paper follows (Weller, Groen, & Hertzberger 1989), testing corroborating sensors before using them for error classification.

(Hanks & Firby 1990) propose a planning architecture suitable for mobile robots. As with (Noreils & Chatila 1995), a plan failure triggers exception handling. The system recovers by either choosing another method randomly whose pre-conditions are currently satisfied (similar in concept to logical sensors (Henderson & Shilcrat 1984) and behaviors (Henderson & Grupen 1990)), or by running the same method again (similar to the retesting strategy used by (Ferrell 1993)). As with (Payton *et al.* 1992), there is no formal error classification scheme. No check is performed to confirm that the sensors providing information about the pre-conditions are still functioning themselves. If they are not, the recovery scheme may pick a method that will either fail, or, more significantly, hallucinate and act incorrectly.

An earlier version of SFX-EH was presented in (Chavez & Murphy 1993). This article builds on that work, with two significant advances. The control scheme is now a global, deliberative process with the ability to access information from sensors not allocated to the behavior. The original was restricted to using only information directly available to the behavior. This was intended to provide fault tolerance entirely within a behavior; in practice with landmark navigation and hall-following this proved to be too severe.

## Approach

This paper concentrates on the exception handling strategy needed to classify a sensing failure. It assumes that an AMR accomplishes a task via independent behaviors which have no knowledge about sensing processes being used by the other behaviors. A behavior is assumed to consist of two parts: a motor process or schema, which defines the pattern of activity for the behavior, and a perceptual process or schema, which supplies the motor process with the necessary perception to guide the next action. This assumption allows the perceptual process to be treated as a logical sensor. Alternative logical sensors may exist for the percept. The sensor and feature extraction algorithms used to compute the percept are referred to as a *description* of the percept, synonymous with a logical sensor. There may be more than one description of a percept using the same sensor. For example, a hazardous waste container can be modeled in terms of 2D visual features or 3D visual features; each set would form a unique description even though they were extracted from the same camera. A logical sensor may fuse the evidence from than one description; this is generally referred to as sensor fusion of multiple logical sensors.

A description is the smallest granularity for identifying a sensing failure; therefore, the difference between a software defect (e.g., the algorithm fails after the 100th iteration) and errant instantiation (e.g., the algorithm is triggered with the wrong parameters) is indistinguishable. However, the exception handler should not assume that a failed logical sensor means that the physical sensor is "bad." Instead, it should attempt to isolate and test the physical sensor separately where possible.

Either the behavior or a global supervisory monitor is assumed to detect a sensing failure and supply the exception handler with the symptom and relevant information. The symptom may provide an explicit classification of the source of the problem, i.e., serves as a complete causal model; for example, upon malfunctioning, the hardware returns a failure code. The symptom may only be a partial causal model (e.g., lack of consensus between observations), thereby necessitating further investigation. The exception handler assumes that there is only one sensing failure at a time. This simplifies classification. By solving one sensing failure, it is hoped that any additional failures would be taken care of. If not, the new logical sensor will fail and exception handling reinvoked. It is worth emphasizing that the system does not assume that any additional sensors used for classification or recovery are operational; therefore any sensors used for corroboration must be validated as functional in advance. Also, the exception handling approach supports graceful degradation by acknowledging when it can't solve the problem and turning control over to whatever mission planning arrangement is used by the robot.

The exception handling strategy is divided into two steps: *error classification* and *error recovery.* The error classification module uses a variation of Generate and Test (Lindsay *et al.* 1980) to generate hypotheses about the underlying cause of the failure. There are three advantages to using Generate and Test. First, since it is an exhaustive search, it catches errors that occur infrequently. Second, Generate and Test allows the robot to actively collect additional information. Because robotic behaviors generally are reactive in the sense of (Brooks 1986), their perception is limited to local representations focused solely on the motor action. As a result, there is usually not enough information available to a behavior to isolate the cause locally. Active acquisition of additional information is critical to the success of error classification. Three, the tests do not require redundant sensors, instead information from other modalities can be used. A Generate and Test strategy does have one disadvantage; because it performs an exhaustive search, it can be time consuming. However, this disadvantage has not been encountered in practice to date because of the small search space for the set of sensors typically used by mobile robots.

Error classification follows the same basic procedure as Generate and Test (Lindsay *et al.* 1980):

1. Generate all possible causes based on the symptom.
2. Order the list of associated tests and execute the tests to confirm any of these causes.
3. Terminate classification when all tests have been performed or an environmental change has been confirmed. Testing does not terminate upon the first confirmed sensor failure because an environmental change can cause a sensor diagnostic test to report a false positive. This can be determined by examining the results of all tests. If the list of tests is exhausted and no source of the failure can be identified, an errant expectation (i.e., planner failure) is assumed to be the cause.

There are five novel extensions to Generate and Test for classifying sensor failures in AMRs. One, the problem space is constrained by the symptom (e.g., missing observation, lack of consensus between multiple observations, highly uncertain evidence, etc.) in order to reduce search. Two, the exception handler generates all possible hypotheses and tests associated with that symptom at one time in order to reduce testing time and resources, and to prevent cycles in testing. Portions of the tests associated with the hypotheses may be redundant; this prevents them from being rerun. Three, the list of hypothetical causes always includes violations of the pre-conditions for each description (sub-logical sensor) in the logical sensor. This is similar in philosophy to GTD (Simmons & Davis 1987) where the debugger challenges the pre-conditions of nodes in the dependency structure. Note that in this application, the challenge is part of the initial hypothesis generation step rather than a debugging step. Example pre-conditions are sufficient ambient light and adequate power supply. Four, the tests are ordered to ensure correctness. If additional sensors are being used in the tests to corroborate observations or verify the condition of the environment, they must first be tested (if possible) to confirm that they are operational. Five, the list of tests is examined and redundant tests removed in order to speed up testing.

Once the sensing failure is classified, recovery is straightforward since the logical sensor scheme explicitly represents equivalences between sensing processes. The search for an alternative degenerates to a table lookup. If the sensing failure is due to either a malfunction or an environmental change, error recovery attempts to replace the logical sensor with an alternative. The alternative must satisfy any new pre-conditions discovered by the classification process. For example, if the reason for a sensing failure with a video camera is because the ambient lighting is extremely low, then a logical sensor using a redundant video camera is not considered. If there is no viable alternative logical sensor, the error recovery process declares a mission failure and passes control to the planner portion of the robot.

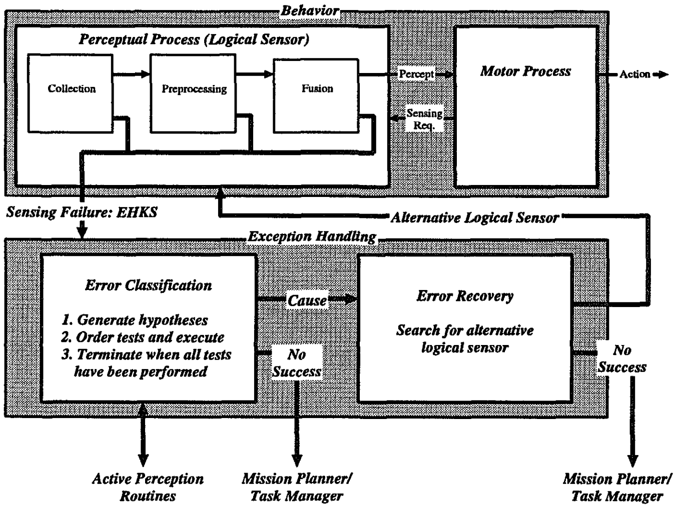


Figure 1: Overview of SFX-EH.

## Implementation: SFX-EH

The exception handling strategy described above has been implemented as an extension to the Sensor Fusion Architecture (SFX) (Murphy & Arkin 1992), called SFX-EH (SFX Exception Handling). Figure 1 shows a conceptual layout of sensing activities in SFX-EH. The perceptual process component of a behavior executes in three steps, as per SFX. First, observations are *collected* from each description in the logical sensor, e.g., grab an image, run feature extraction algorithms on it. Next, the descriptions are *preprocessed* to compensate for asynchronous observations, etc. The *fusion* step integrates the evidence for the percept from each description and passes it to the motor process. Situations where the logical sensor consists of a single description are treated as a degenerate case of sensor fusion and the fusion step is null.

At this time, self-monitoring perceptual processes within a behavior are the only mechanisms for detecting sensing failures, but behavioral and planning level monitoring is not precluded. SFX examines the data for a failure after each step. The four symptoms currently recognized by SFX are: *missing data* (the description has not been updated with a new reading), *highly uncertain data* (the observation of a description is vague or ambiguous), *highly conflicting observations* (the observations from multiple descriptions do not show a consensus), and *below minimum certainty in the percept* (the evidence that the percept is correct is too low for the motor process to safely use). Hardware or dedicated diagnostic software can short-circuit the detection process. If an explicit error is detected, perceptual processing for the behavior is immediately suspended, and the associated recovery scheme implemented (if any) or control is passed to the exception handler.

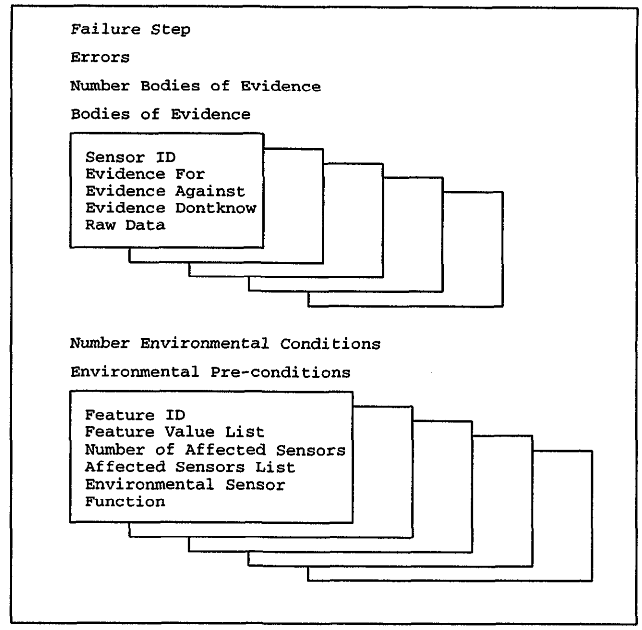


Figure 2: Diagram of the Exception Handling Knowledge Structure (EHKS)

The exception handling module is global. It relies on the Exception Handling Knowledge Structure (EHKS) to provide it with the relevant data about the sensing failure and the task. The EHKS, shown in Figure 2, is a frame with six slots. The **failure step slot** is a flag that describes whether the failure occurred at what stage of execution. The **errors** slot gives the failure condition encountered. The **bodies of evidence** slot is list of frames, each of which holds data from each description in the logical sensor. The **environmental pre-conditions** slot also holds a list of frames, each of which describe the attribute of the environment (if any) which serves as a pre-condition for using that sensor, the expected value of the environmental attribute for acceptable performance of the sensors, and pointers to other sensors which share the same environmental pre-condition. The EKHS contains this so it can challenge the environmental pre-conditions.

The hypotheses take the form that a particular description or logical sensor has failed. The failure conditions describe if the failure occurred during the collection step, the pre-processing step, or the fusion step, along with what type of failure occurred. If the failure occurred during the collection step or the pre-processing step, then individual suspect bodies of evidence are directly known; otherwise, all bodies of evidence are considered suspect.

Once the suspect descriptions have been identified, the actual list of tests is generated. The tests are used to determine the specific cause of the error by investigating potential sensor malfunctions and environmental changes. Generating the test list requires deciding which environmental conditions need to be tested, based on which descriptions are suspect. Because the environmental pre-conditions may hold different attribute values for each sensor, an environmental change can affect some sensors, but not others. Also, because challenging environmental pre-conditions may require additional sensing, the system must be certain that the sensor to be used for additional sensing is operating nominally. Thus, a sensor diagnostic must be run before collecting additional sensor data.

The test list is generated by initially checking if any descriptions in the **Affected Sensors** slot of an environmental frame and in the sensing plan contribute a suspect body of evidence. If so, and no sensing is required to acquire data to determine the value of the desired environmental attribute, then the environmental pre-condition challenge is added to the test list. If additional sensing is required to challenge an environmental pre-condition, then a diagnostic for the sensor which performs the additional sensing is added to the test list *in front of* the environmental pre-condition challenge. Finally, duplicate sensor diagnostic routines are removed from the list, if present. Each test list item contains identification of the test and indicates if the test is for an environmental change, a sensor diagnostic for a sensor contributing a body of evidence, or a sensor diagnostic for an environmental sensor.

## Demonstrations

The current version of SFX-EH has been transferred to *Clementine*, a Denning MRV-4 mobile robot, shown in Figure 3 and demonstrated for landmark navigation using redundant sensors. The objective was to show the operation of the classification and recovery scheme for all types of failures in a realistic setting.

The behavior used for this demonstration was **move-to-goal(goal=purple-square)**, where the goal was a purple square landmark. The behavior was purely reactive; the robot had no a priori knowledge of its relative position. The presence of the landmark in the image elicited a constant attraction potential field. Two logical sensors were available for perceiving the purple square. The default logical sensor consisted of one description taken from the color video camera mounted on front of the robot (camera 0). The landmark was represented by two features: the intensity values in HSV space corresponding to that shade of "purple", and shape via the Hu invariant spatial moments (Pratt 1991). The belief in the landmark was computed as how well the Hu spatial moments of a candidate "purple" region matched the

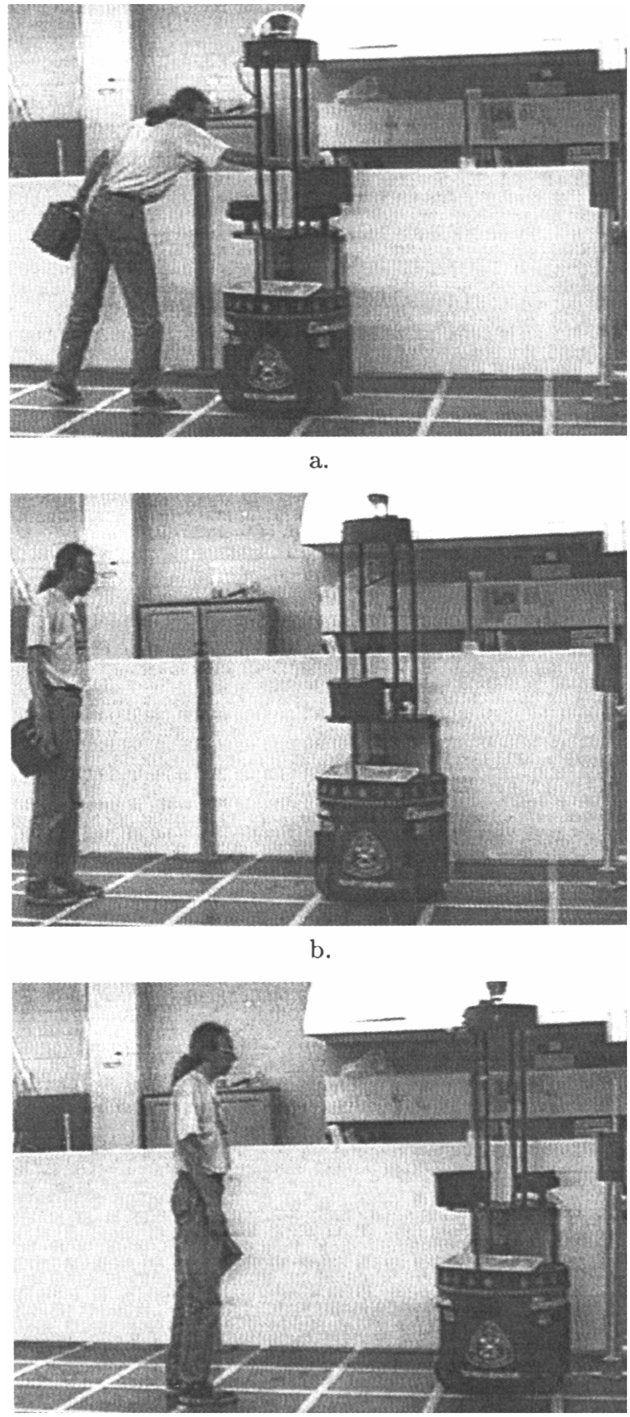


Figure 3: Landmark navigation: a.) Initiating sensor malfunction by covering camera b.) Recovery by turning to alternative sensor (shown in mid turn) c.) Resumption of behavior and completion of task.

```
** STARTING ERROR CLASSIFICATION **
Body of evidence 0: Sensor type is color camcorder

STEP 1: Identification of suspect bodies of evidence
Pre-processing errors discovered
Suspect body of evidence: 0
This BOE did not report missing data

STEP 2: Generation of candidate hypotheses (tests)
Building test list.
        Color camera diagnostic:
        This is an environmental sensor diagnostic.
        Check intensity:
        This test challenges an environmental pre'cond.
        Color camera diagnostic:
        This is a suspect sensor diagnostic for sensor number 0.
        Check intensity:
        This is a suspect sensor diagnostic for sensor number 0.
Done building test list.

STEP 3: Execution of tests

        Test 1: This is a color video hardware diagnostic function.
                to see if any good color cameras exist.
                Testing color sensor number 0 which was marked good...
                Found a good color sensor, number 0. Ok to run other tests.

        Test 2: This is to find out if any color sensor reports good intensity.
                Color sensor 0 reports below minimum intensity threshold.
                Environmental intensity is ok, detected with sensor 1.

== CONFIRMED TEST LIST ======
Color camera error
============================
** ERROR CLASSIFICATION COMPLETE **
** STARTING ERROR RECOVERY ********
Recovery 1
    Original sensing plan:
      description 0: sensor number 0, named Sony- Videocam

Performing color video hardware error recovery.
  REPLACING sensor number 0 with sensor number 1

Repaired sensing plan
Description 0: sensor number 1, named Sony- Videocam
** ERROR RECOVERY COMPLETE **
```

Figure 4: Abbreviated output from SFX-EH.

landmark model. The alternative logical sensor applied the same algorithms but used the color video camera mounted on the rear of the robot (camera 1). While the logical sensors are redundant in terms of the type of information they produce, the robot must face backwards in order to use camera 1 for landmark navigation.

In each run, the robot was placed in an open area within 25 feet of the purple-square landmark. Depending on the purpose of the demonstration, the robot may or may not have been placed facing the landmark. As the robot made progress towards the landmark, a failure would be introduced. Sensor malfunctions were introduced by pulling the video cable out of a camera and putting a box over one camera to simulate a problem with the optics. Turning out the lights, an environmental change, was simulated by putting boxes over both cameras simultaneously. An errant expectation was generated by moving the landmark in the middle of a run or orienting the robot where it was not seeing the landmark.

Figure 3 shows instances from a typical sequence; the corresponding output of SFX-EH is in Figure 4. In Fig. 3a. the robot is making normal progress towards the landmark using the default logical sensor as a graduate student is about to place a box over the camera to simulate a sensor malfunction (e.g., dirt on the lens). In Fig. 3b. the robot has halted while it generates hypotheses and tests them. It uses the video camera in the rear to attempt to establish whether an environmental change has occurred. If so, both cameras should report images with a high average intensity level. The output of the two cameras does not agree; camera 1 shows no indication of an environmental change but camera 0 does. Since the cameras are mounted on the same small robot, it is unlikely that only one camera would be affected by an environmental change. Therefore, SFX-EH concludes that camera 0, or its software, has become defective and must be replaced. Fig. 3c. shows the robot resuming progress towards the landmark, but turned 180° in order to use the alternative logical sensor. The sign of the motor commands are automatically reversed when camera 1 is the "leader;" other behaviors which depend on the direction of motion receive the reversed commands.

## Conclusions and On-going Work

The Generate and Test approach taken by SFX-EH has several advantages. It requires only a partial causal model of sensing failure, and that partial causal model is based on interactions between physical sensors and the environment, rather than limited to models of how the sensors respond for a task, which are difficult to acquire. This is expected to allow the problem solving knowledge associated with a specific physical sensor configuration to be portable to other tasks. The classification process can be short-circuited when all causes of a symptom have the same recovery scheme. The construction of tests takes into account possible confounding from other failed sensors, adding more reliability to the classification and recovery process, plus preventing cycles in testing. Unlike previous systems, the tests themselves can extract information from complementary sensors. The logical sensor organization allows exception handling to be interfaced with the behavioral level of existing robot architectures.

SFX-EH has two disadvantages. The most significant is that the hypotheses and tests are based on domain-dependent knowledge, not purely general purpose problem solving skills. The basic structure can be ported to new applications, but new knowledge will have to be added. However, most of the domain-dependent knowledge is portable because the knowledge base is organized around sensor interactions, not a casual model of the sensors for a specific behavior. For example, a change in the environment can be confirmed with a redundant sensor regardless of what the robot was attempting to perceive prior to the failure. The addition of general problem solving strategies and a learning mechanism, such as

Case-Based Learning, is being considered. Second, the logical sensor representation allows rapid generation of a small set of tests and ease of generation, but introduces other problems due to coarse granularity and a possible lack of available alternate logical sensors. However, these problems, especially the issue of when to re-consider a "bad" physical sensor, appear to be tractable and will be addressed in future refinements of SFX-EH.

The demonstrations provided additional insights and directions for future research. A practical issue is when to retry a sensor that has been identified as "bad." It should also be noted that experience with SFX-EH has shown that the testing, not the problem space search needed for hypothesis generation, is the bottleneck in recovering from a sensing failure. Part of this experience is due to the small set of possible hypotheses implemented at this time. But a large part of the rapid generation of hypotheses is due to a) the category of sensing failure indexing the classifier into the subspace of potentially applicable hypotheses and b) the coarse granularity of the failure modes.

SFX-EH currently lacks the ability to resolve resource contention due to active perception demands and does not update the sensing status to other behaviors. These issues are being actively addressed by the addition of a global event-driven sensing manager. The utility of the SFX-EH style of classification and recovery is not limited to AMR; SFX-EH is currently being applied to intelligent process control for power generation as well.

## Acknowledgments

This research is supported in part by NSF Grant IRI-9320318, ARPA Grant AO#B460, and NASA/JSC Contract NAS.9-19040. The authors would like to thank Greg Chavez for his original research efforts, Elizabeth Nitz for coding a previous version of SFX-EH and the Hu moments, and the anonymous reviewers.

## References

Bajcsy, R. 1988. Active perception. *Proceedings of IEEE* 76(8):886–1005.

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-1(1):1–10.

Chavez, G., and Murphy, R. 1993. Exception handling for sensor fusion. In *SPIE Sensor Fusion VI*, 142–153.

Ferrell, C. 1993. *Robust Agent Control of an Autonomous Robot with Many Sensors and Actuators.* M.S. Dissertation, Department of Electrical Engineering and Computer Science, MIT.

Hanks, S., and Firby, R. 1990. Issues and architectures for planning and execution. In *Proceedings of a Workshop on Innovative Approaches to Planning, Scheduling and Control*, 59–70.

Henderson, T., and Grupen, R. 1990. Logical behaviors. *Journal of Robotic Systems* 7(3):309–336.

Henderson, T., and Shilcrat, E. 1984. Logical sensor systems. *Journal of Robotic Systems* 1(2):169–193.

Howe, A., and Cohen, P. 1990. Responding to environmental change. In *Proceedings of a Workshop on Innovative Approaches to Planning, Scheduling and Control*, 85–92.

Hughes, K. 1993. Sensor confidence in sensor integration tasks: A model for sensor performance measurement. In *SPIE Applications of Artificial Intelligence XI: Machine Vision and Robotics*, 169–193.

Lindsay, R.; Buchanan, E.; Feigenbaum, E.; and Lederberg, J. 1980. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project.* New York: McGraw-Hill.

Murphy, R., and Arkin, R. 1992. Sfx: An architecture for action-oriented sensor fusion. In *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 225–250.

Murphy, R. 1992. *An Architecture for Intelligent Robotic Sensor Fusion.* Ph.D. Dissertation, College of Computing, Georgia Institute of Technology. GIT-ICS-92/42.

Noreils, F., and Chatila, R. 1995. Plan execution monitoring and control architecture for mobile robots. *IEEE Transactions on Robotics and Automation* 11(2):255–266.

Payton, D.; Keirsey, D.; Kimble, D.; Krozel, J.; and Rosenblat, J. 1992. Do whatever works: A robust approach to fault-tolerant autonomous control. *Journal of Applied Intelligence* 2:225–250.

Pratt, W. 1991. *Digital Image Processing, 2nd ed.* New York: Wiley.

Simmons, R., and Davis, R. 1987. Generate test and debug: Combining associational rules and causal models. In *Proceedings of 10th International Joint Conference on Artificial Intelligence*, 1071–1078.

Velde, W. V., and Carignan, C. 1984. Number and placement of control system components considering possible failures. *Journal of Guidance and Control* 7(6):703–709.

Weller, G.; Groen, F.; and Hertzberger, L. 1989. A sensor processing model incorporating error detection and recovery. In Henderson, T., ed., *Traditional and Non-traditional Robotic Sensors.* Maratea, Italy: Springer-Verlag. 225–250.