

Contents

1	Introduction	1
1.1	Major Trends	2
1.2	Criteria for Analysis	3
1.3	Outline	6
2	Common Approaches	6
2.1	Model Based Methods	6
2.2	Expert Systems	7
2.3	Agreement Based Approaches and Formal Analysis	7
2.4	User Interfaces and Health Monitoring	7
3	Summary of Papers	8
3.1	Technology Readiness Level 9	8
3.2	Technology Readiness Level 5	8
3.3	Technology Readiness Level 4	10
3.4	Technology Readiness Level 3	14
3.5	Technology Readiness Level 2	18
3.6	Technology Readiness Level 1	20
4	Conclusions	22
4.1	Major Contributions	22
4.2	State of the Art: Possible Solutions	23
4.2.1	Solution 1: BEAM	24
4.2.2	Solution 2: Rymon and Soika	25
4.3	Conclusion	27

1 Introduction

The aim of this literature review is to explore the state of fault tolerant and health monitoring techniques which can be used for autonomous mobile robots. For the purposes of this review fault tolerance is

defined as the ability of the system to detect and compensate for fault conditions, where those faults may be degradations in performance or a catastrophic failure. Health monitoring is defined as the system's awareness its own resources and other limitations, like sensitivity to heat. A health monitoring control system must also be able to adjust the robot's behavior to reduce or remove any detected health risk.

As robots are asked to perform tasks in more remote and unstructured environments, fault tolerance and health monitoring will become increasingly important features of a successful robotic system. Robots are frequently considered for tasks which are either very remote, for example exploration of Mars, or very dangerous, like nuclear waste cleanup or urban search and rescue [41]. In both cases the importance of the robot's mission lies primarily in its ability to replace humans or other living beings thereby keeping them out of harm's way [25][41]. Sending in a human to repair the robot or to replace a dead battery defeats this purpose. Also, communication over large distances makes it difficult for operators to detect problems in time to correct them [38]. Therefore robots which have the ability to compensate for failures and manage consumable resources on their own will be more suited for these types of tasks.

Guaranteeing a fault-free hardware or software design for unstructured environments is very difficult if not impossible. Unstructured environments, by definition, have many unknown characteristics. This makes them nearly impossible to model for design and testing purposes. Under these circumstances faults become inevitable. Therefore for a robot to successfully carry out its mission in these types of environments it must be able to handle faults in a proactive manner.

There are *two aspects* of autonomous mobile robots which make these features difficult to develop. The first aspect is the complexity and unpredictability of the environment in which these systems must operate. This makes the robot more reliant on its sensor readings and at the same time makes it difficult to predict what those sensor readings should be. The second aspect is limited resources. These systems often have limited power and computational resources and very little redundancy. Therefore any solutions must be as computationally simple as possible and must maximize their ability to use any available resources to fix or replace the broken component.

1.1 Major Trends

Very little work has been published which addresses the problem of fault tolerance for autonomous mobile robots. Therefore the scope of this review has been broadened to include more general diagnostic techniques found in the AI literature. These techniques cover everything from medical diagnosis, to plant

monitoring, to fault tolerance for industrial manipulators and fall into two general trends: model based and those based on expert systems.

The majority of the diagnosis methods found in the general case and also within the robotics literature are *model based*. These techniques rely on the predictions of the data gathered from the sensors in normal and sometimes also in fault modes in order to detect and diagnose problems. In [1] Console shows that model-based diagnosis has reached the level of maturity required for it to be used to solve real world problems, which may explain its prevalence in the literature. Many of these techniques also use learning techniques, trend analysis, and/or agreement based diagnosis. These features are added to reduce the fault tolerant system's dependence on models which may be inaccurate and are prone to errors in novel situations.

Another type of diagnosis method which is less common is based on *expert systems*. These are designed either to encapsulate the knowledge of human experts. These are often designed to work at the symbolic level as aids to doctors or technicians and do not deal directly with the subject or device they are diagnosing. Most of the model based methods, on the other hand, are designed to deal directly with the sensor data from the device. These often also handle detection of faults and are more likely to be implemented and run on the device itself.

Only a few of the diagnosis techniques also handle the problem of finding solutions to the problems discovered. While few have explored the problem of fault tolerance for mobile robots, even fewer have considered the problem of health monitoring. None of the papers found investigate health monitoring as a goal in and of itself.

1.2 Criteria for Analysis

In order to properly analyze these papers it is necessary to define this problem in terms of specific features which an ideal solution must provide, thereby establishing a clear goal.

A *fault tolerant* system must provide the robot with an awareness of:

- *The state of its sensors.* Autonomous robots rely on accurate sensor data in order to function in an open world. Therefore they must be able to detect when a sensor is not functioning or when a sensor is malfunctioning and to what extent it is malfunctioning.

- *The state of its effectors.* Robots depend on their effectors to navigate and interact with the world around them. They must be able to detect when a motor or other effector is not functioning or malfunctioning and to what extent.
- *Environmental changes which cause sensors to give inaccurate percepts.* Sensors can be working properly but due to environmental changes may not be supplying accurate percepts. The resulting problems are often very similar to those encountered with faulty sensors, and are just as debilitating to the robot's performance.
- *Environmental changes which cause motors to or other effectors to malfunction.* Like sensors, effectors may be working properly but due to environmental changes their ability to interact with that environment may be impeded. An example would be track slippage. Again these faults usually manifest themselves in similar ways to effector faults and are just as crippling.
- *Environmental changes which cause errant expectations.* Changes in the environment can also cause planning level problems. These faults are the result of a discrepancy between the current state of the environment and the planning routine's expectations, or model of the world. These inaccurate models may be built from flawed information, like maps, given to the robot in advance. Flaws can also come from more complex perceptual schemas which, while providing accurate information about the environment, may have missed an important detail because it took too much time to update the model.

A successful fault tolerant system must be able to not only detect each of these problems but must also be able to distinguish between them. This isolation is as important as it is nontrivial since several causes may have the same effect but require different solutions. For example, consider a broken camera versus someone turning off the lights. If the camera is broken it may be possible to switch to a working one. If the problem is actually that the ambient light in the environment is too low then the second camera will be just as useless as the first, therefore another solution is required [10].

While in theory any system which meets the above requirements will provide a robot with the fault tolerance it needs, a successful implementation of a fault tolerant system has several more criteria it must meet. For a fault tolerant system to be useful it must reside on the robot itself and therefore it must compete with other processes or behaviors for limited resources. Therefore any solution must not

use too much memory for storing and processing the required information and as stated earlier must be as computationally simple as possible. An ideal system would also alert the robot's operator of any problems which will lead to a noticeable reduction in the robot's performance.

A system which also provides the robot with the capability to monitor its own health must supply another set of features.

A *health monitoring* system must provide the robot with an awareness of:

- *Battery level.* Battery power is an important resource for mobile robots therefore a health monitoring system must be aware of the battery level and the energy requirements of all its components.
- *CPU utilization.* Another important resource is processor time. In order to more effectively use this resource the robot needs to be aware of processor utilization as well as have a general idea of the CPU load and update rate of the software routines which will help it to complete its task.
- *Communications.* Many robots communicate with their operators through wireless connections therefore the robot must be made aware of communications drop outs. It must also be aware of the signal strength as some types of connections will draw more power to boost a weak signal.
- *Shock force,* for example of an impact. The robot must be aware of any shock force and the tolerance of its components to withstand this type of force.
- *Temperature.* The robot must also be aware of the ambient temperature and the low and high tolerance of its components to that environmental factor. Other components of the robot like the battery, motors, and electronics (CPU, motor controllers) generate their own heat. These require different actions to be taken to prevent overheating and therefore need to be monitored separately.

Only a robot which is provided with this awareness can assess its own health condition. While that assessment is very important, it is also important that the robot be capable of responding to any perceived health risks. Therefore any health monitoring system must also be able to use this information to select more efficient behaviors or adjust the parameters of behaviors to conserve the robot's resources. In extreme cases it should also be able to activate corrective behaviors like those used by animals to maintain homeostasis, for example moving to a cooler area to prevent overheating. A successful health

monitoring system must be capable of informing the robot's operator of impending problems. It should also allow the operator to override health-aware decisions, or force the robot to put itself at risk if necessary.

1.3 Outline

First the papers included in this review will be broken down into general groups based on commonalities in their approach to the problem of fault detection and diagnosis. Then each will be briefly summarized in order by rank. Each paper is ranked based on NASA's Technology Readiness Levels often referred to as TRL [21]. TRL provides a common metric which can be used to compare the maturity levels of different types of technology. This metric particularly useful for this analysis since few of the techniques explored in this review were designed for the same application.

2 Common Approaches

This section begins by breaking the papers down into general groups based on commonalities in their approach to the problem of fault detection and diagnosis.

2.1 Model Based Methods

Model based diagnosis methods use models of the target system to predict the correct values for input data. These predictions are compared to real data from the target system in order to detect and isolate potential faults.

The majority of these systems rely on rigorous models of the projected input data. Among these are [2], [4], [18], [22], [27], [33], [38], [39], and [40]. Mackey's system [19] can be considered to be in this group due to the fact that it can work with a complete model, though it is not required. Rinner's system [30] can also be placed in this group though it is distinctive in its ability to take a weaker model and improve over time.

Other model based methods use less rigid models. Some of these techniques work at the symbolic level, using only casual or partial causal models of the target system. These are [3], [6], [7], and [37]. Two of the methods found [5] and [12] use more localized models relating one set of sensor's data to another, as opposed to modeling the system itself. [11] takes a similar approach but uses more complex, trained

models to detect and isolate faults in sensor-rich systems. Another pair of papers describe techniques which model the target system at a more deliberative or planning level, these are [14] and [36].

2.2 Expert Systems

Diagnosis techniques which take the expert system approach are more interested in capturing the diagnosis process used by human experts, rather than capturing an accurate model of the target system or the input data itself. These methods tend to be less common than the model based methods. Despite this fact, the techniques found have been successful in a variety of applications and tend to be more accessible from a user's perspective. These include [9], [13], [20], [29], and [32].

2.3 Agreement Based Approaches and Formal Analysis

Only two of the techniques found do not rely on models of either the data or the diagnostic process. These methods instead rely upon agreement between several working redundant sensors in order to detect faults. These are [17] and [35].

There are several papers included in this review which do not present diagnosis or fault tolerant techniques. Instead they present *formal models* for the analysis and design of such systems. There are [8], [28], and [34]. For an even more general view of how such systems can be designed Stefik's work [31] is also included. It describes the cost structure of sensemaking which is the process of finding the best representation and encoding data in that representation to answer task-specific questions.

2.4 User Interfaces and Health Monitoring

While [9], [13], and [19] all spent some design time improving their interfaces, there are two papers included in this review whose entire goal is to improve communications between robots and their operators. Both [15] and [16] present natural language systems used to describe a robot's behavior to their programmers or operators.

Only two techniques have been found to date which cover anything close to the health monitoring requirements outlined in section 1.2. In [23] Michaud presents a control system which allows a robot to autonomously recharge itself and in [26] Naik presents strategies for improving the energy efficiency of software.

3 Summary of Papers

Each paper will be briefly summarized in order from the most to the least mature based on NASA's Technology Readiness Levels (TRL). This metric ranks technologies on a scale of 1 to 9, 1 being basic research and 9 being proven fielded technology. The application of this metric in this paper follows the model found in the TRL White Paper [21] except that many of these technologies are not being developed for application in space. Therefore their maturity level was based instead on their application to the target environment indicated by the paper. Papers which fall into the same TRL category are summarized in alphabetical order within that subsection. It should also be noted that the category is determined by the level of maturity of the technology at the time the paper was published.

3.1 Technology Readiness Level 9

A technology reaches TRL 9 when it has been used continually in its target environment and the last of the "bug fixing" tasks is complete. At this point it has established its ability to solve the problem for which it was developed and has established its reliability in its target environment. Only one paper in this review describes technology which has reached this level.

In [9] Helfman presents an expert diagnostic system which has actually been fielded by the US Army. It presents a high level description of Turbine Engine Diagnostics (TED) and discusses why this application was successful. The system includes diagnostic, maintenance, bookkeeping, and training modules plus a visual inventory of the engine's parts. The reasoning system which studies the resulting structure is procedural or goal-oriented. TED was the first fielded maintenance system in the field of AI. Though little detail is provided in this paper on the system itself, there is plenty of evidence that this system was successful and that its design can be used for similar field diagnosis systems. One possible limit to its applicability is that it seems to be designed solely as a mechanic's aid, no mention of interfaces between the engine itself and the system are described.

3.2 Technology Readiness Level 5

A technology reaches TRL 5 when it is validated in a less controlled environment, similar to the actual target environment for that technology. Any other components of the complete system, in which this technology must function, must at least be simulated in order to provide an accurate testbed. This level

is distinguished from TRL 6 by the fact that at TRL 6 a working prototype of the complete system must be developed and the system must be tested and proven to be successful in the actual target environment. There are four papers in this review which describe technologies at this level.

In [14] Lamine presents a monitoring system based on temporal fuzzy logic for use with behavior-based robots (also discussed in the author's previous work). The general approach taken in the system is built upon a temporal logic system developed for monitoring distributed real-time systems. The authors have incorporated fuzzy semantics into a similar system to make it more suitable for a robot working in an open world. The truth value of a proposition is based on several snapshots over a set period of time, which is determined empirically. An ordered weighted average (OWA) operator is used to evaluate the truth value over time. In the noise elimination section classical noise removal filters for salt and pepper and Gaussian noise are simply placed in the OWA operator. The system was implemented to collect data and alert an existing robot control program of any detected failures. Results were obtained by running the program through three sets of tests in a real world environment, the first set of tests served as a base line. On the second set of tests the system detected 22 failures which did occur. The third set of tests included the noise filtering which resulted in 25% fewer failures, all of which were detected. This technique is useful because it provides a flexible framework in which to define faults in the system including planning and navigation problems, but seems to be very computationally complex.

In [17] Lee presents a temperature measurement system which uses an array of 36 identical sensors and a self-diagnostic algorithm to find faulty sensors in the array. The voltage from each of the sensors is compared to a constant reference voltage, producing a binary output. This algorithm assumes that the majority of the sensors provide accurate readings, therefore the sensors with the minority state (0 or 1) are suspected of being faulty. Experimental results showed that using the self-diagnosis algorithm to remove faulty sensors allowed the system to continue producing accurate readings when up to a sixth of the sensors were malfunctioning. This method depends on a higher level of redundancy than is usually found on robot platforms.

In [23] Michaud presents an emotion-based control system which allows a robot to autonomously recharge itself. Emotions are used to monitor the progress of the robot's goals or motives over time. The value of the motive called Energize is used to determine if the robot needs to recharge and for how long to recharge. There are three factors which influence this value: the battery voltage level, detected presence of the charging station, and the rational module. The rational module may reduce the value of

Energize so that the robot does not try to recharge in the middle of a critical task, or increase the value to encourage the robot to recharge before a long task. Each motive has a priority which combined with its value determines to what extent that motive influences the robot's actual behavior. This system has been implemented for a set of Pioneer 2 robots one of which was used at the AAAI 2000 Mobile Robot Challenge. Two have also been used in experiments designed to see if the emotional control scheme helps them to share a recharging station efficiently. The recharging behavior was effective in all these scenarios. The advantages of this system are that it uses a general model for autonomous recharging which improves its portability, allows for opportunistic recharging, and takes the robot's current and future responsibilities into account.

In [35] Soika presents a failure detection framework based on probabilistic analysis of correlation between redundant sensor readings. According to the paper this method does not require explicit failure models. It is also designed to work independently of the type of failure and the robot's environment. Conditional probabilities, similar to those often used for sensor fusion, are used to determine the consistency of a sensor reading in relation to the information provided by the other sensors. An example application of this framework using an occupancy grid representation is presented. The application was tested on an autonomous mobile robot called ROAMER turning in an open environment at 20 degrees per second. An inconsistency grid was shown for two sensors along with the combined occupancy grid. It is not clear from the paper how many of these experiments were carried out or how the data was evaluated. As this technique builds up belief over time it is a more robust solution than other agreement based techniques which consider only one point in time. Considering that this system does not rely on models or training, does not care whether the sensors are physically or logically redundant, and takes into account that redundant sensor readings should agree it would be interesting to see how reliable this framework can be for sensor fusion based robots.

3.3 Technology Readiness Level 4

A technology reaches TRL 4 when it is validated in a laboratory or simulated environment. Technology at this level goes beyond "proof-of-concept" work to begin exploring the details of how it can be implemented to solve the problem for which it was developed. Seven technologies covered in this review fall into this category.

In [13] Krishnamurthi presents a system which generates expert systems for machine fault diagnosis.

The development of the system is based on studies which showed that the same basic processes and data structures are used by technicians to diagnose many different classes of machine. General diagnosis techniques are captured in built-in diagnosis modules which consists of a shallow reasoning, a deep reasoning module, and a learning component which records the results of the deep reasoning module. A device can be defined at any level of abstraction of the real system. For validation the program was used to create an expert system for diagnosing a Cincinnati Milacron 786 robot. The paper stated that a large number of diagnostic scenarios were tested and validated by a human expert though no data was presented to support this claim. One limit to its applicability is that it is designed to generate diagnostic aids for human users, it is not designed to deal directly with the devices it is diagnosing.

In [18] Lerner presents a fault detection system for complex dynamic systems. They present a novel method grounded in a combination of existing techniques, namely temporal causal graphs (TCG's) and Kalman filters. The interactions between components of the system are described using dynamic Bayesian networks which are derived from a TCG of the system and are expressive enough to capture temporal dependencies as well as discrete and continuous data. A technique similar to the extended Kalman filter is then used to estimate the current state of the system as the set of "beliefs" that the system is in each state. Finally the future state of the system is predicted which generates more accurate "belief" values. Experimental data was gathered from a model of a system which contains five water tanks and only three measurement points. Using only the measurements the diagnosis system was able to accurately detect faults in the tank system. The obvious drawback of this method is that it requires a very rigorous model of the target system. It is also difficult to tell if the system can detect faults in real time since the paper specifies how long it took to detect a fault in terms of steps as opposed to runtime.

In [19] Mackey presents an overview of an extensive architecture for failure prediction, detection, and isolation along with performance metrics. The system can handle many types of input from discrete status variables to real valued sensor data. The data is first filtered to remove any information which is deterministically known from the supplied system model. Analysis is then done on the remaining sensor data in groups (correlation-based) and individually (feature-based). Both analysis components determine fault conditions based on deviations from learned normal states and their respective data models. Another component is used to determine if the two analysis components agree on the presence of a fault from a specific source, if so the system has confirmed that the fault exists. This component also looks for any disparities between software execution (symbolic data) and hardware operation (sensor

data). Two modules are used to predict problems. One predicts trends in the short term to determine if any limit values will be exceeded. Another tracks performance in long term trends to predict more subtle problems and to report on system performance. Yet another component is used to compile all of this information for use by a planning system or human user. This architecture has been used in several application domains the results of which are found in 5 papers cited. A similar system designed for autonomous robots would provide all the functionality required for fault tolerance, but that development would take a fair amount of time and data gathering to develop models of the target systems.

In [26] Naik presents three strategies for improving the energy efficiency of software at the implementation level. The study focused on comparing recursive versus iterative implementations, different implementations of the same algorithm, and different algorithms with the same complexity which solve the same problem. The energy cost of each implementation was determined by implementing it in C code and adding up the energy cost of each instruction of the resulting compiled assembly code. The energy cost was based on the current (in milliamps) required for the instructions on an Intel 486DX2. Several commonly used algorithms were used in the analysis. The three general strategies found include: assigning live variables to registers, avoiding repetitive address computations, and minimizing memory accesses. Another important finding is that the constant factor behind the asymptotic complexity is also an important factor in energy cost. Experimental results showed energy savings from 18% to 60% were possible using these strategies. Robots are a very energy conscious application area therefore these strategies are of use, and in some cases are already being used.

In [29] Reed presents a diagnosis method that correctly identifies multiple defects, even when those defects interact. A recognition-based reasoning module is trained from an existing knowledge base. This module is used to determine what important symptoms could be present. A hypothesis construction module then combines those symptoms into a set of hypotheses based on heuristics which are also learned from the knowledge base. Hypotheses are evaluated based on the ratio of explained symptoms to unexplained symptoms. The system was tested with data from the medical files of 78 children with heart problems. The test results showed that the system performed almost as well as the experts for cases with single, complex, and multiple defects present. Though the domain this system was implemented for was simplistic, only 7 single defects were considered, the approach is general enough to be applied to any domain. This technique may be particularly useful for diagnosing more challenging cases when failures occur together. Other advantages are that it does not require a model of the system and is sufficiently

fast for online diagnosis. The difficulty in using this system will be in gathering adequate training data.

In [32] Rymon presents a system for assisting physicians in the treatment of patients with multiple traumas. The system is goal based in that it does not care about finding the source of the problem, but instead focuses on the steps required to return the subject (patient) to a normal state. The system decides which action should be taken next based on a set of beliefs, attitudes which is a measure of the relevance of information, and a set of rules which encapsulate expert knowledge. Beliefs are formed from observations provided by the physician and anything that can be concluded from the rules. Attitudes are formed from the rules alone. The planning system can also use the rules to determine if the subject has the resources available to perform the actions required and in what order to perform multiple actions. In experiments the system was provided with information from 97 real cases. The paper states that a panel of three trauma surgeons blindly preferred the actions recommended by the system to the actions which were actually carried out in the majority of cases. Even though this system was designed for medical diagnosis, this is the most promising method which relies on an expert diagnosis system rather than a model of the system. Its primary advantage is that it is completely goal-oriented and that it considers the limited resources required to probe for the diagnosis and to repair of the problem in its decision-making process. Some major modifications and additions would have to be made to interface this system with the sensor data on an online system.

In [39] Vos presents a fault tolerant control system for autonomous unmanned air or underwater systems with logical redundancies. The goal of the paper is to reduce the problem of controlling UAV's or UUV's which are inherently linear parameter dependant (LPD) systems into a problem which has already been solved, namely control of linear time invariant (LTI) systems. Using feedback linearization techniques and a coordinate transform LPD systems can be transformed into a set of coordinates where the parameters do not change. This allows the designers of a fault tolerant system to develop one model which can be used to compare to the actual state of the robot over the entire range of its operating envelope. Significant deviations from this model are treated as faults and the source is isolated. This process in turn causes the system to reconfigure its control laws to compensate for the problem. The results given in the paper are based on three experiments where the control system was given telemetry data from flights of an experimental UAV. In two of those cases faults caused the UAV to crash in the real world whereas the control system was able to recover the vehicle in simulation. In a third the flight did not suffer from any faults and the control system's commands followed the actual system's commands exactly.

While the results were good the need for a precise model makes this approach difficult to implement for autonomous robots.

3.4 Technology Readiness Level 3

A technology reaches TRL 3 when the underlying concepts and applications found in TRL 2 have been proven in a laboratory-based study. At this level the experiments only have to provide a “proof-of-concept” for the target application. Experimental proof or detailed analysis is not required. Eleven technologies covered in this review fall into this category.

In [4] Deuker presents a neuro-symbolic hybrid system for diagnosis of faults in unmanned underwater vehicles. According to the paper, the key advantage of this system is that it can learn from failures not modeled in its initial set of rules. Other advantages are easier translation of the diagnosis process into a form that the operator can understand and fewer examples required for a usable system. Two examples were given to show the usefulness of the system, both conducted using a simulator of an experimental underwater robot. The first set of results involve a diagnosis already modeled in the initial rules, whereas the second set show the system’s ability to learn from a new failure. This technique is useful where a rigorous model of the normal functioning of the robot is available for fault detection and there is ample time to train the system. Both of these resources are hard to come by in the domain of unmanned ground vehicle self-diagnosis.

In [6] el Ayab presents a neural network design which models a cause-to-effect reasoning process. The network is designed around the goals of finding the simplest and most probable (most supported by the evidence) solution. The network works at the symbolic level where nodes are identifiable causes and effects. Causes with the same effect compete with each other until only one remains, where the ones with the highest initial value are favored. Results are presented from analysis of a simplified partial causal model of a car. While the results are convincing the method assumes that each effect has only one cause which does not hold for most autonomous robotics applications. Though it might be a good method for offline validation of existing models.

In [7] Feret presents a formal definition of Experience-Aided Diagnosis (EAD). EAD combines model-based and case-based reasoning. Model-based reasoning requires at least a partial causal model of the system. Weaknesses in the model are covered by case-based reasoning, here the diagnosis system will save any cases where the model produced an incorrect diagnosis. In this manner the system learns

information which is either missing from the model or modeled incorrectly. The paper does not explicitly cover how to combine the two methods, stating that this part of the process is domain dependant. Though no experimental results are included in this paper, two other papers by the authors are cited which include implementations of this model with results. The applicability of this type of diagnosis as an enhancement to purely model-based methods is obvious. A better system would also be capable of updating the incorrect or incomplete model. Like any other learning technique some training time will be required before the system is serviceable depending on the accuracy of the model.

In [11] Hung presents a diagnostic system which uses a combination of existing signal processing and reasoning techniques. It is designed to improve fault detection and analysis for systems with large amounts of data coming in from similar and/or distinct sets of sensors. Two methods are mentioned for pulling significant information out of the raw data signal: short-time Fourier transform which considers time slices of a specified length, and wavelet analysis. Since both methods can produce results in high-dimensional parameter spaces, Principle Component Analysis (PCA) is used to reduce the dimensionality without losing the most important information from the previous analysis. Then a composite feature is created by finding the product of all the feature spaces of the individual sensors. Bayesian decision theory is then used to combine data over time in the composite feature space. The conditional probability density function for each hypothesis is determined from the training data. It is also used to in two different metrics: one used to determine the correct hypothesis, and the other to determine the distance between the state of the target system and its normal state (Mahalanobis distance). Results from experiments, from two different configurations of the system, show that these distance measurements reduce the problem of detection and diagnosis to a simple thresholding operation. By using a combination of proven methods the paper develops a system which is general and works well, but may be too computationally complex to run on a robot.

In [15] Laegle presents a natural language system for explaining autonomous error recovery for mobile robots. The goal of this paper is to increase the accessibility of robots to their operators. The natural language system was added to the interface for the KAMRO robot which is an autonomous, mobile, assembly robot built to replace the industrial manipulators often used in factories for assembly tasks. The system uses model-based error detection, isolation, and recovery techniques. These techniques are focused on updating the modeled world state and then continuing the task with the updated information. The abstract model used for error recovery breaks the problem down into the following elements: operation

where the error occurred, error, cause, error recovery operation, and high level goal of the error recovery plan. These elements are related to sets of nouns and verbs which can be used to create coherent verbalizations of that element. These verbalizations are then placed in text frames to create natural language sentences. Advantages of this system include dialog based interaction where the user can request as little or as much information as they want and the use of a general fault model which improves its portability.

In [16] Le presents a natural language system for explaining a behavior-based robot's actions. The goal of this paper is to increase the transparency of robots to their programmers and operators. The system is designed to respond to questions like "Why are you [symptom]?". It answers these questions by following an abstract casual circuit from the [symptom] to its cause. Abstract casual circuits are made up of two types of elements. The first type are operators or the set of behaviors which could be running at any time. The second type are propositions or conditions which can be true or false and can activate or deactivate operators. The user can request information about either operators or propositions, which allows for follow up questions. The system was implemented and tested on a modified iRobot Magellan robot. Preliminary results are good though more work is needed to create a fully functional interface. This system will provide a good tool for debugging robot software. It will also provide the robot with the capability to teach operators how it works in an natural, interactive manner.

In [20] Madden presents a system which generates fault trees from databases of classified sensor data and then automatically creates a monitoring and fault diagnosis program from the results. The tree is then converted into rules in C language code and linked into a shell which is also in C. Experimental sensor data was gathered from a simulation of a rigorously tested model of a pneumatic servo-controlled robot arm. The results in general are poor, due to insufficient normal data points. Other problems are also acknowledged including the fact that the success of the diagnosis is heavily dependant upon the presence of similar examples in the training set. This weakness makes this system more useful for the industrial manipulator considered then for autonomous robots.

In [27] Narasimham mainly covers exactly the same system as in [22] with a small contribution from [11]. The only concept presented in this paper which is not covered in the other two is the notion of model-driven adaptive signal processing. The model considered in [22] is expanded to include knowledge of the best form of signal processing to use in order to detect problems in the mode related to that model.

In [30] Rinner presents a method for monitoring hybrid (continuous and discrete) systems. It looks

for significant discrepancies between the current mode's model and the trends in the observed data while simultaneously refining that model. Significant discrepancies are assumed to be transitions to other modes. It also tracks multiple hypotheses in parallel where each hypothesis is made up of a set of modes in sequence. These hypotheses therefore make up a high-level model of system behavior either in normal or fault conditions. Experimental results were obtained from a simulated model of a two tank system with and without error. The advantages of this method is that only weak models are required to initialize the system and it can handle asynchronous data updates. It was not clear from the paper whether or not the refined models are sufficiently accurate to monitor a system.

In [36] Stuck presents a system for detecting navigational mistakes made by mobile robots in open environments. The focus of this paper is to consider global mistakes which lead the robot down incorrect paths as opposed to simple local errors. It also places more emphasis on visual mistakes like misrecognition. The system finds mistakes by generating expectations based on a priori information and past results. If the current sensor readings significantly differ from expectations then the "conviction" level of the robot drops below a threshold and the robot stops. At this point it analyzes the data recorded so far and generates hypotheses as to what the mistake was and when it occurred. Heuristics are used to rank the resulting hypotheses. The system was implemented and tested in a simulated environment. A mistake was correctly detected in all eight experiments. In five of the experiments the robot ranked the actual mistake as the most likely hypothesis. The correct hypothesis was ranked no lower than third. The paper acknowledges that the simulated environment allowed them to simplify the vision and motor problems to an extent not possible in a real environment. The usefulness of this technique lies in its ability to detect high level navigational mistakes which lower level diagnosis systems would miss. As this technique builds up belief over time it is a more robust solution than other techniques which consider only one point in time.

In [38] Visinsky presents a layered fault tolerance system for robots. The low and middle layers maintain mathematical models of the dynamics of the system. The lowest layer looks for small discrepancies between the model and the sensor data and is designed to correct any such biases. The middle layer uses a collection of independent tests to monitor sensor data and isolate any errors. If an error is found the sensor or motor involved is replaced by a redundant entity in that subsystem. If the middle layer cannot find a redundant entity it fails up. The top, or supervisor, layer uses an expert system to perform more advanced fault tolerance. It tracks the overall state of the robot and the failure rates of various parts of

the robot. It is also aware of alternatives for fault recovery which are not available in the middle layer. This layer periodically checks to see if the current goal is still reachable in the degraded state of the robot. This system was tested using a simulated model of two different robotic arms. Results showed that the system was able to compensate for simulated failures and complete its task safely. Only one set of results is shown, it is not clear from the paper if more experiments were conducted. This technique's layered approach makes it more suitable for hybrid robotics architectures. Unfortunately its reliance in the low and middle layers on accurate models of normal sensor data make it harder to apply to autonomous robots. The top layer's capabilities including replacement between subsystems for malfunctioning components, tracking the failure rates of components, and tracking the overall capabilities of the robot to complete its task are rare but needed features of fault tolerant robotics systems.

3.5 Technology Readiness Level 2

A technology reaches TRL 2 when the basic understanding of the technology has reached a level at which practical applications can be identified. This level is usually considered to still be within the domain of basic research and is often carried out by the research rather than the engineering community. Five technologies covered in this review fall into this category.

In [5] Djath presents a control system for a mobile robot with multiple sensors which are physically or logically redundant. Faults are detected in two ways. First, if the sensor gives a reading outside of the possible range of predicted readings. Second, if the sensor's state (faulty or not faulty determined using the first method) does not fit with the state of the other sensors then that sensor is considered to be faulty. Once a fault is detected the control logic is modified to accommodate the change. The paper seems to concentrate more on using the state of the sensors (again faulty or not faulty) to help determine the environment of the robot then on developing a robust determination of that state.

In [12] Ishida presents a distributed diagnostics system modeled on the human immune system. In this system each sensor has its own agent which is constantly comparing that sensor's outputs with others in the same network. Each agent acts independently, communicating dynamically and in parallel with the other agents. These networks can be built for a set of sensors which provide redundant information even if the set of sensors involved are not homogenous. This method was the advantage of being modular which fits well into a behavioral framework, but the high level of overhead makes it a less practical solution to the robot diagnosis problem.

In [22] McIlraith presents continued work by the authors on diagnosing hybrid (discrete and continuous) systems. The methods in the paper therefore build on previous work done by the authors and it does not include any experimental results. Faults are detected simply by significant deviations from their expected values, though more sophisticated methods of failure detection have been developed by this group. Temporal causal graphs (TCG's) are used to model the interactions between components of the system. These graphs are used to generate a set of candidate hypotheses for the deviation through backward propagation. Two processes run in parallel for each candidate hypothesis. The first is a qualitative-based process which propagates the proposed fault forward through the TCG. If the results do not match the observations then that candidate is thrown out. The second process performs quantitative model fitting to get more precise measurements of the time the failure occurred and any parameters associated with that error (used to determine the severity). Comparison of the final candidates is performed based on their optimal fitness to the observations. Two drawbacks are that this method requires rigorous models of the target system and that the authors doubt that it can work in real time. Though their plans for future work address this issue.

In [33] Sary presents a new technique for processing telemetry from spacecraft systems. This technique augments existing trend analysis methods with model, case, and rule-based reasoning. Faults are detected as deviances from the normal trend for each component or subsystem in the model. Rule-based reasoning is used to diagnose and solve all well known anomalies. For new anomalies case-based reasoning is used to determine if similar anomalies are already modeled. Information about anomalies which have been encountered before but are not well known are modeled using Local Dempster-Shafer theory. The model provides an ordered set of hypotheses to test against the telemetry. A similar process is used to find solutions once the diagnosis is complete. This work is ongoing therefore no experimental results were included in the paper. Though this system is model based it has two advantages: it is designed to work in real time and has an online learning capability.

In [40] Washington presents a preliminary attempt to create a fault detection system for rovers. The system is based on a combination of Markov models and Kalman filters. Markov models are used to model the qualitative discrete states of the robot. Kalman filters use continuous data from the sensors to estimate the state of the robot to be used by the Markov models. The system detects faults when the combined system concludes that the robot is in a fault state. Experimental tests were primitive in nature and designed to simply show the need for further development of this system. All the models and

parameters needed were determined “by hand” whereas in a mature system they would be based on real data and experience with the platform. Despite this fact the results, taken from 50 different experiments, showed that the system could correctly identify faults in the majority of cases. The prototype system was optimized at multiple points leading to an update rate of under a second for all 6 wheels in sequence on a platform comparable to the resources onboard the rovers. Though the run time and reliability of this system are advantageous the time it would take to build reliable models is a disadvantage of this system.

3.6 Technology Readiness Level 1

A technology reaches TRL 1 as soon as the basic principles of that technology are understood and articulated. Seven papers in this review fall into this category. Included in this set are formal studies of fault tolerant systems. These papers do not develop any particular technology but attempt to further the understanding of these types of systems in general.

In [2] Darwiche presents a formal study of how other factors, such as device behavior, can be used to improve purely structural diagnosis techniques. The process used to actually find the fault is treated as a black box in this study, therefore it does not present any practical methods for performing diagnosis except at a very high level. Experimental results were taken from randomly generated devices consisting of only buffers and “and/or” gates. If a connection or structural based diagnosis system is implemented, it would be useful to look at the improvements presented here. Otherwise this information is not very applicable to the problem of robot self-diagnosis.

In [8] Goldberg develops a formal model of collaborative control systems where sources are modeled as finite automata. The goal is to formally explain numerous reports which suggest that such systems are highly fault tolerant. Sources in this system could be multiple sensors, multiple control processes, or multiple human operators. The control signal is calculated by averaging the signals from each of the sources. Experimental data from a simulator consisting of 100 sources trying to trace a circle is presented in graph form. It shows that performance is actually improved when sources send no signal or the signal is inverted. In the next section these results are mathematically proven to be correct.

In [3] Klerer presents an extension to an active diagnostic system which considers the cost trade-off between probing and finding the next best hypothesis to test. The system uses a model of the device based on constraints and an assumption-based truth maintenance system to generate hypotheses.

The original system only performed tests on components based on the hypothesis which best fits the information gathered so far. The new information is then used to reevaluate all the hypotheses until only one remains. The best hypothesis thereby eliminates an unknown percentage of the current search space, whereas a probe can be selected which will always eliminate half. The new system is given the cost per probe versus the cost per second of computation and uses these metrics on each update to determine whether to skip the search for the best hypothesis and start probing. Experimental results were evaluated based on the computed cost. The new system shows significant improvements over the older system. Though this system does not deal with varying costs per probe it is not difficult to see how that feature could be added. Provided that the slight computational overhead does not cause any issues a similar extension could be used to optimize an existing robot diagnostics system.

In [28] Perraju presents an extended version of I/O automata specifically designed to model mission critical systems. The goal is to extend the capabilities of these automata so they can be used to capture normal versus fault or recovery actions taken by the system. Another key requirement is that timing issues such as periodic actions and deadlines can also be captured in the new automata. A model was created of a fire control system of a combat vehicle in order to test the expressiveness of the new automata. The model was able to capture the fault tolerant requirements of the system.

In [34] Sheldon presents a new method for analysis of fault tolerant systems using a static task graph technique. This technique makes it easy to identify key components of a system design and their dependencies. Formal techniques are described for converting these graphics into a set of probabilities which can be used to compare the relative reliability of the target systems. Experimental results were presented based on a comparison of three different models for a fault tolerant subsystem. Several graphs were presented showing the predicted reliability of the system over varying reliability of its components. The strengths and weaknesses of each model could be clearly determined from these graphs.

In [31] Stefik presents an analysis of the cost structure for sensemaking tasks. Sensemaking is defined as the process of searching for a representation and encoding data in that representation to answer task-specific questions. The most important finding in this analysis is that the process of making sense of a complex body of information always follows a common pattern made up of similar operations and cyclic processes. This is supported by flow diagrams for four unrelated sensemaking tasks each of which followed the same basic pattern. The contribution of this analysis therefore lies in a deeper understanding of the common elements of these types of tasks and the subsequent improvement of interfaces for human and

automated agents based on that knowledge. A deeper understanding of the cost structure of sensemaking can also lead to better defined scopes of such projects. Any successful fault tolerance or health monitoring technique for a system as inherently complex and resource deprived as autonomous mobile robotics will have to select the most important information to be stored and the best possible representation for that information. Therefore the process of sensemaking will be an important part of any attempt to solve these problems.

In [37] ten Teije presents a formal study of approximate diagnosis using subsets of casual models to find the correct hypothesis. Four general strategies are presented for resolving the problem of too few or too many hypotheses or to adjust the size of hypotheses. Each of these strategies works by adding positive or negative observations to the subset of the model to be considered. The key advantage of this approach is that elements of the model are added incrementally and the system often does not have to consider the entire model in order to find a solution. Though the strategies are described using a simplified behavior model of a car as an example, no experimental results are included in the paper. If a good casual model can be developed for the faults encountered on mobile robots these strategies might be useful in optimizing the diagnosis system.

4 Conclusions

4.1 Major Contributions

Mackey's [19] system provides an example of a mature fault detection, isolation, and prediction system which uses a combination of qualitative and quantitative methods. This system is highly adaptive in that it has been used for various applications with little modification and is also capable of adjustable autonomy. The major contribution of this effort lies in the proof that such a flexible, complex system can be scaled to run on systems with slow processors and limited resources and still serve its purpose. Another contribution is the use of a "gray box" approach to modeling the target system. In a "gray box" approach the system adapts to the completeness of the supplied model. Both of these contributions are important to the problem of fault tolerance for autonomous mobile robots in that limited resources and incomplete models are key obstacles in this domain.

Stefik's work on sensemaking provides a major contribution by supplying a better understanding of the process which leads to the type of models (physical or trained) which have made systems like

Mackey's successful. By isolating common operations and patterns of reasoning followed by professional sensemakers this work has laid the foundation for streamlining and eventually automating that process.

Rymon [32] and Reed [29] have provided solutions to particularly difficult problems in the area of medical diagnosis without the use of models of the complex physiological systems involved. Using only encoded expert knowledge or training data these systems were able to meet or surpass the diagnosis capabilities of the doctor who treated the patients. The major contribution of this effort lies in the proof that the knowledge of human diagnosis experts can be captured and used effectively without the need to model the target system itself. These also provide solutions which are goal-oriented and use probing or active sensing, which is a capability also available to robots, to reach the correct diagnosis.

Soika [35] and Lamine [14] both developed fault detection and diagnosis methods specifically for behavior-based autonomous robots. Soika's technique is of particular interest in that it does not require any a priori knowledge of the robot or its environment. The major contribution of this effort is in applying a proven method for sensor fusion in robots to the problem of fault detection and isolation. The top layer of Visinsky's [38] system also provides some important contributions to the field of fault tolerance for robots. That layer's capabilities including tracking the failure rates of components and tracking the overall capabilities of the robot to complete its task are rare but needed features of fault tolerant robotics systems.

The technique developed by Michaud [23] appears to be the only method in existence which can help meet some of the health monitoring requirements outlined in section 1.2. The autonomous recharging behavior outlined in the paper is driven by an adroit combination of percepts which can be found on any behavior-based robot platform. Though the charging station was designed around the robot's needs and had to be clearly marked for the robot to find it, this technique presents a solid first step in the direction of full autonomy for mobile robots.

4.2 State of the Art: Possible Solutions

Correlating the requirements for fault tolerance discussed in section 1.2 with the capabilities of each technique covered in this review yields two possible solutions to the problem of fault tolerance for autonomous mobile robots, each with its own advantages and drawbacks.

4.2.1 Solution 1: BEAM

With some minor modifications Mackey's [19] system design, called BEAM or Beacon-based Exception Analysis for Multimissions, could be used to implement a fault detection and diagnosis system for an autonomous mobile robot.

This solution would provide an autonomous mobile robot with an awareness of:

- *The state of its sensors.* BEAM uses a two components to detect and isolate sensor faults.
- *The state of its effectors.* One component of BEAM is dedicated to detecting discrepancies between the system's software execution and the hardware operation, which leads to detection of problems with any effectors on the robot.
- *Environmental changes which cause sensors to give inaccurate percepts.* Though BEAM does not explicitly support the isolation of environmentally induced rather than sensor error induced faults, there may be a way to impart this distinction into the system's models.
- *Environmental changes which cause motors to or other effectors to malfunction.* This capability may be added in the same way that environmentally induced sensor faults could be detected.
- *Environmental changes which cause errant expectations.* The same component of this architecture which can isolate hardware problems could also conceivably detect these types of problems as well.

The advantages of this solution are:

- All the components of the resulting system are already designed to work together.
- It is designed to interface directly with the target system as well as the operator.
- Two components provide short term and long term predictions of future fault states. These could be used to give an operator or control system sufficient time to remedy the problem.
- Robotics applications of this design have been developed.

The disadvantages of this solution are:

- The design only handles fault detection and isolation. Recovery methods are expected to be handled by either the operator or the device's control system.

- The application specific aspects of the design will have to be worked out and those solutions may leave the implemented system in a less portable state.
- BEAM is large and complex. It will take a significant investment of time to develop all the required components and the resulting system may entail too much overhead to reside on a mobile robot without impeding its performance.
- The design does not use probing or active sensing which is possible in the domain of diagnosis for autonomous mobile robots.

4.2.2 Solution 2: Rymon and Soika

A combination between the diagnosis expert system described by Rymon in [32] and the fault detection and isolation method described by Soika in [35], with some minor modifications could provide a complete fault tolerance system for an autonomous mobile robot.

This solution would provide an autonomous mobile robot with an awareness of:

- *The state of its sensors.* Soika's system is designed to detect and isolate sensor faults.
- *The state of its effectors.* Using the command sent to the motor controller as another source of data, Soika's system could also be used to detect these types of faults.
- *Environmental changes which cause sensors to give inaccurate percepts.* Like sensor faults these types of errors will most likely be caught by Soika's system or the perceptual schema itself. Isolating these errors is possible through probing [10] which can often establish if the equipment is working or not and Rymon's system relies on probing.
- *Environmental changes which cause motors to or other effectors to malfunction.* Again these errors are often distinguished from motor faults through probing which Rymon's system provides.
- *Environmental changes which cause errant expectations.* Further experimentation with Rymon's expert system may provide a means of dealing with these types of faults as well. If not then a model based method like Mackey's [19] or Stuck's [36] may have to be added at the planning level.

The advantages of this solution are:

- Soika's system is explicitly designed for behavior based robots and is based on a technique which has already shown its usefulness in this domain.
- Rymon's system is goal-oriented which provides a proactive recovery system.
- Rymon's system also considers the fact that probing and recovery procedures require the use of limited resources and schedules these procedures based on those constraints.
- This system would be highly portable. Soika's detection technique is designed for portability. Rymon's expert system would be based on human experts' techniques which are known to be portable. Any required adaptations or additions required (like adapting motor controller commands to be sent to Soika's detector) can reside outside of the fault tolerant control system and use a common interface.
- Rymon's expert system is designed to model human intelligence and is therefore more suited for human interaction.
- Both Soika's and Rymon's techniques are of relatively low computational complexity.

The disadvantages of this solution are:

- Rymon's system was designed for medical diagnosis, not robotic diagnosis.
- An appropriate means of interfacing the various components of this system will have to be developed.
- It is not clear how early Soika's system can reliably detect an anomaly. Sufficient warning of an impending fault may not be possible with this system.

Michaud's system [23] is an excellent first step in meeting the health monitoring requirements outlined in section 1.2. Otherwise the problem of health monitoring for mechanical systems like autonomous robots has not been covered in the literature. A good starting point for research in solutions to this problem may be found in the homeostasis systems found in animals.

4.3 Conclusion

The success of robots in tasks such as urban search and rescue, military reconnaissance, and exploration of the solar system will continue to depend on the robot's ability to recover from faults and efficiently use the resources available to it. Even before autonomous robots were considered for these tasks the need for fault tolerance and health monitoring for robots was established [24] but very little progress has actually been made. This is especially true for robots which must function in an open world, ironically the robots which need these capabilities the most. Drawing from efforts made in other application areas may help those who are developing robotic solutions for these challenging domains made the necessary steps to develop fault tolerant, health-aware robotic systems.

References

- [1] L. Console and O. Dressler. Model-based diagnosis in the real world: Lessons learned and challenges remaining. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1393–1400, 1999.
- [2] A. Darwiche. Utilizing device behavior in structure-based diagnosis. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1096–1101, 1999.
- [3] J. de Kleer and O. Raiman. Trading off the costs of inference vs. probing in diagnosis. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1736–1741, 1995.
- [4] B. Deuker, M. Perrier, and B. Amy. Fault-diagnosis of subsea robots using neuro-symbolic hybrid systems. In *IEEE Oceanic Engineering Society. OCEANS'98. Conference Proceedings*, pages 830–834, 1998.
- [5] K. Djath, M. Dufaut, and D. Wolf. Mobile robot multisensor reconfiguration. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, pages 110–115, 2000.
- [6] B. el Ayeb and Shengrui Wang. Computing effect-to-cause/cause-to-effect diagnoses within ndl. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1332–1338, 1993.

- [7] M. Feret and J. Glasgow. Experience-aided diagnosis for complex devices. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 29–35, 1994.
- [8] K. Goldberg and B. Chen. Collaborative control of robot motion: Robustness to error. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium*, pages 655–660, 2001.
- [9] R. Helfman, E. Baur, J. Dumer, T. Hanratty, and H. Ingham. Turbine engine diagnostics (ted): An expert diagnostic system for the m1 abrams turbine engine. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 1032–1038, 1998.
- [10] David L. Hershberger. Mobile robot sensor fault recovery improvements. Master’s thesis, Colorado School of Mines, June 1997.
- [11] E. Hung and F. Zhao. Diagnostic information processing for sensor-rich distributed systems. In *Proceedings of the 2nd International Conference on Information Fusion (Fusion 99)*, 1999.
- [12] Y. Ishida. Active diagnosis by self-organization: An approach by the immune network metaphor. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 1084–1089, 1997.
- [13] M. Krishnamurthi and D.T. Phillips. An expert system framework for machine fault diagnosis. *Computers and Industrial Engineering*, 22(1):67–84, 1992.
- [14] K.B. Lamine and F. Kabanza. History checking of temporal fuzzy logic formulas for monitoring behavior-based mobile robots. In *Proceedings 12th IEEE Internationals Conference on Tools with Artificial Intelligence*, pages 312–319, 2000.
- [15] T. Langle, T.C. Luth, E. Stopp, and G. Herzog. Natural language access to intelligent robots: Explaining automatic error recovery. *Artificial Intelligence: Methodology, Systems, and Applications*, pages 259–267, 1996.
- [16] Christopher Dac Le. System transparency through self-explanation. November 2002.

- [17] Choon Fatt Lee and Yong Ping Xu. A multi-sensor based temperature measuring system with self-diagnosis. In *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON 2001*, pages 903–906, 2001.
- [18] U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *Proceedings Seventeenth National Conference on Artificial Intelligence AAAI-2000*, pages 531–537, 2000.
- [19] R. Mackey, M. James, Han Park, and M. Zak. Beam: Technology for autonomous self-analysis. In *2001 IEEE Aerospace Conference Proceedings*, pages 2989–3001, 2001.
- [20] M.G.M. Madden and P.J. Nolan. Monitoring and diagnosis of multiple incipient faults using fault tree induction. In *IEE Proceedings-Control Theory and Applications*, pages 204–212, 1999.
- [21] John C. Mankins. Technology readiness levels. White Paper 028, NASA, NASA Headquarters Washington, DC 20546-0001, April 1995.
- [22] S. McIlraith, G. Biswas, D. Clancy, and V. Gupta. Towards diagnosing hybrid systems. In *Hybrid Systems and AI: Modeling Analysis and Control of Discrete Plus Continuous Systems. Papers from the 1999 AAAI Symposium*, pages 124–131, 1999.
- [23] Francois Michaud and Jonathan Audet. Using motives and artificial emotions for long-term activity of an autonomous robot. In *Proceedings of the fifth international conference on Autonomous agents*, pages 188–189. ACM Press, 2001.
- [24] Glenis Moore. Robot programming. the language of labour? *Electronics & Power*, July 85, 1985.
- [25] Robin R. Murphy. *Introduction to AI Robotics*, chapter 6, pages 195–256. The MIT Press, 2000.
- [26] Kshirasagar Naik and David S.L. Wei. Software implementation strategies for power-conscious systems. *Mobile Networks and Applications*, 6(3):291–305, 2001.
- [27] S. Narasimham, F. Zhao, G. Biswas, and E. Hung. Fault isolation in hybrid systems combining model based diagnosis and signal processing. In *Proceedings of the 4th Symposium on Fault Detection, Supervision and Safety for Technical Processes (Safeprocess 2000)*, pages 512–517, 2000.

- [28] T.S. Perraju, S.P. Rana, and S.P. Sarkar. Specifying fault tolerance in mission critical systems. In *IEEE High-Assurance Systems Engineering Workshop Proceedings*, pages 24–31, 1997.
- [29] N.E. Reed. Constructing the correct diagnosis when symptoms disappear. In *Proceedings Fifteenth National Conference on Artificial Intelligence AAAI-98*, pages 151–156, 1998.
- [30] B. Rinner and B. Kuipers. Monitoring piecewise continuous behaviors by refining trackers and their models. In *Hybrid Systems and AI: Modeling Analysis and Control of Discrete Plus Continuous Systems. Papers from the 1999 AAAI Symposium*, pages 164–169, 1999.
- [31] D.M. Russell, M.J. Stefik, P. Pirolli, and S.K. Card. The cost structure of sensemaking. In *Proceedings of INTERCHI '93. Human Factors in Computing Systems*, pages 269–276, 1993.
- [32] R. Rymon. Goal-directed diagnosis-diagnostic reasoning in exploratory-corrective domains. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1488–1493, 1993.
- [33] C. Sary, C. Peterson, J. Rowe, T. Ames, K. Mueller, W. Truskowski, and N. Ziyad. Trend analysis for spacecraft systems using multimodal reasoning. In *Multimodal Reasoning. Papers from the 1998 AAAI Symposium*, pages 157–162, 1998.
- [34] F.T. Sheldon, H. Mei, and S.-M. Yang. Reliability prediction of distributed embedded fault-tolerant systems. In *Fourth International Symposium on Software Reliability Engineering Proceedings*, pages 92–102, 1993.
- [35] M. Soika. A sensor failure detection framework for autonomous mobile robots. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovative Robotics for Real-World Applications*, pages 1735–1740, 1997.
- [36] E.R. Stuck. Detecting and diagnosing navigational mistakes. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, pages 41–46, 1995.
- [37] A. ten Teije and F. van Harmelen. Exploiting domain knowledge for approximate diagnosis. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 454–459, 1997.

- [38] M.L. Visinsky, J.R. Cavallaro, and I.D. Walker. A dynamic fault tolerance framework for remote robots. *IEEE Transactions on Robotics and Automation*, 11(4):477–490, 1995.
- [39] D.W. Vos and B. Motazed. The application of fault tolerant controls to uavs. In *Proceedings of the SPIE - The International Society for Optical Engineering*, pages 69–75, 1996.
- [40] R. Washington. On-board real-time state and fault identification for rovers. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, pages 1175–1181, 2000.
- [41] C.R. Weisbin and G. Rodriguez. Surface systems r&d in nasa’s planetary exploration program. *IEEE Robotics & Automation Magazine*, pages 25–34, December 2000.