A Survey about Faults of Robots Used in RoboCup

Gerald Steinbauer

Institute for Software Technology
Graz University of Technology, Graz, Austria
steinbauer@ist.tugraz.at

Abstract. Faults that occur in an autonomous robot system negatively affect its dependability. The aim of truly dependable and autonomous systems requires that one has to deal with these faults in some way. In order to be able to do this efficiently one has to have information on the nature of these faults. Very few studies on this topic have been conducted so far. In this paper we present results of a survey on faults of autonomous robots we conducted in the context of RoboCup. The major contribution of this paper is twofold. First we present an adapted fault taxonomy suitable for autonomous robots. Second we give information on the nature, the relevance and impact of faults in robot systems that are beneficial for researcher dealing with fault mitigation and management in autonomous systems.

1 Introduction

A major goal of research in Robotics and Artificial Intelligence is to develop truly autonomous robots that are able to assist humans in exhausting or repetitive tasks or in dangerous environments. Examples include factories, mining, search and rescue missions and space exploration. In such application domains dependability plays a major role because autonomous robots must not endanger people sharing common space, face unexpected situations or allow no human intervention in the case of problem. In the context of autonomous robots dependability is strongly coupled with fault mitigation.

Autonomous robots in academia and industry frequently show faults or undesired behaviors. This is caused by physical faults in the robot's hardware, bugs in the robot's software, unreliable algorithms or unexpected situations not considered during development. A lot of research has been conducted to develop automated methods for fault-tolerance, diagnosis and repair to be able to cope with these problems. Such methods improved the reliability of robots but still we see a large fraction of robots used in academia or industry fail due to the reasons mentioned above.

Although, the topics of dependability are very important surprisingly there are few founded publications and studies about the reliability of autonomous robots and the relation between faults, counter-measures and dependability. Such work either qualitative or quantitative seems yet to be extremely interesting for all autonomous robots researchers and developers. So far we have only few information about what the common types of faults and problems are. Therefore, no answer can be given to the question what issue tackled had the most positive impact on robot system faults. It might be nice if one has a method that allows a robot to automatically cope with a particular type of fault. If this type of fault, though, occurs once a year in one particular robot the methods has very limited impact on the general reliability of autonomous robots.

X. Chen et al. (Eds.): RoboCup 2012, LNAI 7500, pp. 344–355, 2013.

Based on the above observations we investigated the question: *What are the common, probable and most critical faults and malfunctions of autonomous robots?* In order to give a first founded answer RoboCup is an ideal environment. RoboCup is one of the leading initiatives to foster research and development in Artificial Intelligence and Robotics [1]. In RoboCup a large number of different research robots are used that demonstrate a wide variety of system architectures and properties. Moreover, the tasks and environments used in the RoboCup competitions are dynamic and complex and ask for non-trivial solutions. In order to be able to identify the common and therefore interesting faults to deal with we initiated a survey on robot faults in RoboCup. In the survey groups participating in different RoboCup leagues had been asked to describe their robot systems, the type and frequency of faults their systems suffer from and the impact of these faults on the mission success.

This primarily qualitative survey provides information about the properties of faults of robots used in RoboCup. It can be used to derive common activities and processes to improve the robot's reliability and to guide research into promising directions. For sure this survey is also interesting for researchers and developers outside the RoboCup community as the robots and techniques used in RoboCup are state of the art and similar robots are used in other fields of research and industry as well.

2 Definitions and Related Work

In this section we give basic definitions that are used throughout the remainder of the paper.

A *failure* is an event that occurs when the delivered service deviates from correct service. An *error* is that part of the system state that can cause a subsequent failure. A *fault* is the adjudged or hypothesized cause of an error.

For the current survey on faults of robots used in RoboCup we adapt the fault taxonomy of [2]. Our taxonomy shown in Figure 1 is more suitable for this particular domain. The category *hardware* comprises all faults related to physical faults of the

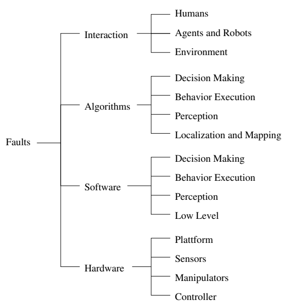

Fig. 1. Fault taxonomy used in current survey

robot's equipment. Design and implementation faults of the robot's software systems fall in the group *software*. The category *algorithms* aggregates problems of methods and algorithms based on its nature. Here we distinguish between a faulty implementation of an algorithm and shortcomings of the algorithm itself. In the category *interaction* we group problems that arise from uncertainties in the interaction with the environment (perception and acting), other agents and humans.

Furthermore, we use several attributes to classify faults and their properties. Some of them are reused from the literature others are newly defined or clarified. The used attributes are: (1) *relevance* of a fault for different robot systems, (2) relation to *condition* and events in the environment, (3) relation to *symptoms* (failures), (4) *impact* to the mission's success and (5) *frequency* of the occurrence of a fault.

We choose the fault categories and attributes based on our experience in research for fault diagnosis and repair [3,4]. Basically we found out that classical faults in hardware and software are not sufficient explanations for undesired behavior of an autonomous robot, e.g., the non-determinism of actions in the real world has a big influence on dependability. Moreover, much researcher work on advanced solutions to handle particular faults but ignore the question whether the fault is very much relevant. Therefore, information on the properties of faults such as impact is important.

As mentioned in the introduction there is few literature and studies on faults of autonomous robot systems. One of the few examples is [2] where the authors did a qualitative and quantitative study on faults of robots and their impact in the search and rescue domain. Moreover, probabilities were given for how likely it is that parts fail. In [5] the authors presented an autonomous tour guide robot and described its continuous operation for five month. The authors reported a Mean Time Between Failures (MTBF) of around 4.6 hours of the robot system and noted that most of the failures were caused by software components. This is one of the few long-term studies.

Intuitively one may assume that robot systems having abilities to automatically recover from faults will show a higher dependability. The authors of [6] showed that a team of robots with cooperative repair capabilities has a better chance to complete a cooperative mission. Research on automated recovery from faults in the context of autonomous robots systems is a very active field. The techniques comprise automated detection, identification and repair of faults. If the former two techniques are combined we refer to it as diagnosis. Diagnosis and repair techniques are quite heterogeneous in their properties (qualitative, quantitative or hybrid) and are able to cope with a wide range of faults (hardware, software or knowledge). In [7] the authors showed how particle filter can be used to detect faults in the mobility unit of a planetary rover. Many approaches like [8] specifically addressed the diagnosis of sensing and/or actuator faults. In [3] the authors presented a framework for detection and repair of faults in the control system of autonomous robots. The framework was based on qualitative model-based diagnosis.

To improve the dependability of autonomous systems a number of approaches have been proposed to handle unexpected situations caused by the environment. For instance there is a lot of work in the area of planning and execution monitoring to handle execution faults and unexpected external events. Example architectures are LAAS [9], CLARAty [10] and the Livingstone architecture by Williams and colleagues [11].

3 Investigated RoboCup Leagues

The robot platforms, the methods and the task used in RoboCup resemble to a high extent what is used and done around the world in Robotics and AI labs. In order to have access to a wide variety of robots we included the following leagues in our study: Middle Size, Rescue Robot and RoboCup@Home.

In the *Middle Size League* (MSL) two teams of up to six autonomous robots each play soccer against each other [12]. This league is the one that approximates real soccer best in terms of speed of the game and size of field (18 m width). The used robots have only some size and weight restrictions and move with a speed of up to 5 m/s. No external sensors or interventions are allowed. Research challenges in this league comprise methods for controlling a robot at high speed, localization and navigation with sparse landmarks or machine learning for behaviors like dribbling and team coordination.

The *Rescue Robot League* (RRL) [13] provides a testbed for robots in the urban search and rescue domain. The goal is to develop robots that are able to explore hazard sites after disasters like earth quakes or chemical accidents without endangering first responders. In the competition teams have to explore a given arena built of standard test modules such as ramps, step fields or stairs. The two major tasks are to localize trapped victims and to build a map of the arena. Research challenges comprise mechanical design of robust agile robot platforms, simultaneous localization and mapping (SLAM), exploration strategies or object recognition for victims and hazard material signs.

In the RoboCup@Home League (@Home) [14] service robots for office and home environments are developed. The test bed is an apartment with common parts such as a kitchen, a bed room or a living room. During the competition the robots have to autonomously perform several predefined tests such as to guide a guest to a particular place, find a given object or recognize persons. These resemble simple useful tasks in daily life. Moreover, teams can show individual achievements like newly integrated sensors or special capabilities of their robot in a free challenge. Research challenges comprise safe navigation in indoor environments, grasping and manipulation of objects, human-robot interaction (HRI) or object and face recognition.

4 Survey and Data Collection

4.1 Information about Fault and the Questionnaire

The aim of our work is to get a qualitative estimation which parts of robot systems are affected by faults, what the root causes are, what the symptoms (failures) caused are and what their impact and frequency is. These parameters very much matters for the design and implementation of fault diagnosis and repair approaches.

In order to collect the data needed to answer the above questions we divide the survey into nine main categories:[1]

1. general information on the groups and their hardware and software
2. the robot platform and related faults

[1] The full questionnaire, the full data set of the survey and further interpretations of the data are publicly available at `http:\\www.ist.tugraz.at/rfs`.

3. the robot's sensors and related faults
4. the robot's manipulators and related faults
5. the robot's control hardware and related faults
6. the robot's software and related faults
7. the used algorithms and related faults
8. off-line and on-line measures to deal with faults
9. research focus and general feedback

The questions in categories 1 and 9 provide general context information about the used robots and the research groups involved. Category 8 collects information about fault mitigation strategies already used in the field. The aim of the questions in the categories 2 to 7 is to collect broad feedback on possible faults in various robot subsystems and their impact. Therefore, these categories are further divided in five standard blocks:

1. how much are the related parts affected by faults
2. what are the dominant root causes for faults
3. how much fault are correlated with root causes
4. what are dominant symptoms (failures) caused
5. what is the impact of faults on the mission success rate
6. what is the frequency of particular faults

For instance the related parts of the robot's platform comprise among others wheels, tracks, motors, motor drivers or batteries. Root causes comprise physical problems such as wear or damage, connection or configuration problems or environmental conditions. Symptoms related to faults comprise immobility, power failures, unpredictable behavior or missing data. The selection has been done based on our experience in the construction of autonomous robots, the participation in RoboCup and the daily lab work as well.

The impact of faults is categorized as *not critical, repairable/compensable or terminal*. The frequency of faults are categorized as *never, sporadic, regularly or frequently*. These categories are related to the schedule of the RoboCup competition. *Repairable* denotes faults that can be repaired during a mission or game while *terminal* denotes faults that lead to an abortion of a mission or robots not able to take part in a game anymore. *frequently* means that a fault occurs in each mission or game at least once.

The questions about the relevance of faults and how much parts are affected by faults had to be answered by integers from 0 (not relevant or affected) to 5 (very much relevant or affected). For the correlation of faults to causes matrices were used where the entries represented a particular correlation and could be checked during answering.

4.2 Data Collection Process

The data collection process was conducted as on-line survey using a renowned commercial survey platform. We invited 68 research groups from 21 countries that participated in the last years in the RoboCup world championship or a major regional competition to complete the survey. 16 of these groups participated in Middle Size, 22 in Rescue Robot and 30 in RoboCup@Home. The survey was available on-line from August to November 2010. We received 25 responses to the invitation. This represents a good number of 36,7 % of the invited groups. Due to incomplete submissions we incorporate only 17 responses (25 % of the invited groups) in our survey. 6 responses came from the Middle Size, 5 from the Rescue Robot and 6 from the RoboCup@Home league. A response rate of 25 % of useful responses is above the usual rates in surveys and can be explained by the compact group of researchers participating in RoboCup and the interest in empirical data.

5 Results and Interpretation

5.1 General Information

We start with general information about the used robot systems. Table 1 left shows an overview of the used robot platforms. Nearly 80 % of the research groups use custom-built robot. This fact, though, is not equally true for all application areas. While groups in MSL and RRL mainly use custom-built robot platforms, @Home groups use commercially available platforms to a high extent. For the locomotion of the robot platform mainly differential drives (35 % of the groups), omni-directional drives (35 %), skid drives (25 %) or tracks and flippers (18 %) are used. Please note that some groups use more than one platform or locomotion type.

Table 1. Robot platforms (left) and sensors (right) used by teams in the different RoboCup leagues. Some teams use multiple platform types. Number of teams.

Platform	MSL		RRL		@Home		Overall	
	#	%	#	%	#	%	#	%
Pioneer/GuiaBot	0	0	1	20,0	2	33,33	3	17,65
Telemax	0	0	0,33	6,7	0	0	1	5,88
Madilda	0	0	0,33	6,7	0	0	1	5,88
Volksbot	0	0	0	0	1	16,67	1	5,88
other commercial	0	0	0	0	2	33,33	2	11,76
custom-built	6	100	4,34	86,6	1	16,67	12	70,59
sum	6	100	5	100	6	100	17	100

Sensor	MSL		RRL		@Home		Overall	
	#	%	#	%	#	%	#	%
Odometry/Encoder	6	100	5	100	5	83,3	16	94,1
Directed Camera	3	50	4	80	4	66,7	11	64,7
Hokuyo LRF	0	0	4	80	5	83,3	9	52,9
Omni-Camera	6	100	1	20	0	0	7	41,2
Sick LRF	0	0	0	0	5	83,3	5	29,4
Compass	2	33,3	3	60	0	0	5	29,4
IMU	1	16,7	3	60	0	0	4	23,5

Table 1 right shows an overview of sensors used on the robot systems. Some of them as odometry and directed cameras are commonly used in all domains. Others are used frequently in particular domains such as the light-weighted Hokuyo Laser Range Finder (LRF) (only RRL and @Home) or omni-directional cameras (mainly MSL). Specific sensors such as gas sensors are used very seldom for special purposes only.

Further used hardware components are manipulators. All groups except one from RRL and @Home use such devices. The most common types are the Katana robotics arm and custom-built solutions.

Notebooks are used as central control unit by over 60 % of the groups. Industrial PCs follow with 23,5 %. The major interfaces to hardware used are USB (by 88,2 % of the groups), RS232 and Ethernet (64,7 % respectively) and IEEE139a (52,9 %).

5.2 Robot Platform

We continue with results about faults of the robot platforms. Table 2 left depicts the feedback to which extent the various parts are affected by faults. Columns 2 to 7 shows

Table 2. Ranking of parts of the robot platform affected by faults (left) and the relevance of causes (right)

Platform Part	Ranking (0 .. not affected. 5 .. much affected)						
	0	1	2	3	4	5	avg.
	%						
Batteries	29,4	11,8	41,2	5,9	11,8	0	1,59
Motor Drivers	33,3	20,0	26,7	6,7	13,3	0	1,47
Controller Boards	29,4	41,2	11,8	5,9	11,8	0	1,29
Tracks	62,5	0	12,5	12,5	0	12,5	1,25
Motors	46,7	33,3	0	0	20,0	0	1,13
Gears	66,7	6,7	13,3	0	13,3	0	0,87
Wheels	68,8	6,3	12,5	6,3	6,3	0	0,75
Flipper	75,0	12,5	0	0	0	12,5	0,75
Chassis	75,0	18,8	0	0	0	6,3	0,5

Causes	Ranking (0 .. not relevant. 5 .. much relevant)						
	0	1	2	3	4	5	avg.
	%						
Connectors	11,8	5,9	23,5	17,6	23,5	17,6	2,88
Communication	17,6	17,6	23,5	17,6	11,8	11,8	2,24
Physical Impact	31,5	12,5	25,0	12,5	6,3	12,5	1,88
Wear	35,3	5,9	29,4	17,6	5,9	5,9	1,71
Vibration	29,4	17,6	23,5	17,6	11,8	0	1,65
Damage	23,5	41,2	11,8	11,8	5,9	5,9	1,53
Configuration	35,3	23,5	23,5	5,9	5,9	5,9	1,41
Temperature (Overheat)	52,9	5,9	11,8	11,8	11,8	5,9	1,41
Short Circuits	37,5	25,0	18,8	12,5	0	6,3	1,31
Environmental Conditions	76,5	5,9	11,8	5,9	0	0	0,47

the fraction of groups that assigned the corresponding scores (row 2) to various faults (column 1). Column 8 depicts the average score. It shows that batteries and motor driver are the primarily affected parts. An average score of 1,59 and 1,47 shows that these parts are moderately affected. Anyhow, some parts like tracks or flippers got individual high scores showing that some groups have serious problems with faults in these parts.

Furthermore, we were interested in the primary causes for faults in the robot platforms. Table 2 right depicts the relevance of different causes. The major causes for faults of robot platforms are problems with connectors (average score of 2,88) and communication problems (e.g., protocol or transmission, score 2,24). Moreover, we asked for the relation of faults and causes. According to the response connector problems are major causes for faults of batteries (40 %), controller boards (43 %) and motor drivers (42 %). The numbers reflects the fraction of groups that see a relation between causes and faults. Another prominent cause is wear with relation to faults of gears (75 %) and batteries (50 %).

Another interesting aspect is to which symptoms (failures) the different faults lead. Usually the correlation between symptoms and root cause is not obvious. With the same relevance score of 0 to 5 the major symptoms are immobility (average score of 3,12), unpredictable behavior (2,67) and reduced controllability (2,56).

Finally, we asked for feedback on the impact and frequency of faults of parts of robot platforms. Faults of batteries, motors and controller boards are the main reason for a termination of missions. Most of the other faults were mainly classified as repairable. Faults of batteries and motor drivers occur sporadic to regularly. The other faults occur never to sporadic.

5.3 Sensors

We used the same set of questions and scores for feedback on sensors used by the groups. According to their feedback the major sensors affected by faults are Hokuyo LRF (average score of 1,78), Swiss rangers (1,25), directed cameras (1,23) and sonars (1,2). IMUs (score 0,6) and odometry (0,65) are the most reliable sensors.

The most relevant causes for sensors faults are connectors (average score of 2,73), configuration problems (2,54) and communication problems (2,5). The most significant relation between sensor faults and causes exist between the Hokuyo LFR and configuration problems (43 %) as well as physical impact (43 %). More than 50 % of the groups mentioned a relation between most of the sensors and connector problems.

The major symptoms (failures) caused by sensor faults are that the sensor delivers no data (average score of 3,76) and wrong or corrupted data (2,47). An unstable data rate is a minor symptom (0,75). According to the responses faults of Sick LRF are terminal to the mission while most of the other faults are repairable. Faults of directed cameras and sonar occur regularly while most other faults are classified as sporadic.

5.4 Manipulators

Manipulators seem to be sensitive parts. They received a high average score of 3,09 for being affected by faults. The leading causes for faults are physical impact (average score of 3,7), damage (3,42), communication problems (2,17) and connectors (2,0). Major symptoms (failures) are problems with the kinematics (average score of 2,88), the precision (2,55) and the payload (2,33). Faults of manipulators have also high impact on the mission success as 45,4 % of the groups classified them as terminal. Nevertheless, over 70 % of the groups classified manipulator problems as sporadic.

5.5 Control System

This section only concerns control system's hardware such as computer and notebooks. The major causes for faults of the control system are connectors (average score of 2,24 out of 5,0), configuration problems (2,0), communication problems (2,0) and vibrations (1,36). The most prominent symptom of control system faults is that peripheral functions (e.g. particular ports) are missing or non-functional (average score of 2,88). This symptom is followed by the fact that the system does not boot (1,71), hangs (1,6) or crashes (1,6). Most of the symptoms such as the system does not boot, reboots or hangs were classified by over 50 % of the groups as terminal. The responses show that most problems with the control hardware are serious but occur only sporadically. But one third of the groups replied that missing functionality occurs regularly.

5.6 Robot Software

This section concerns software engineering aspects of the robot control software. We asked for faults of 13 major software parts such as computer vision, self-localization, low-level drivers, decision making, inter-process communication or behavior execution.

According to the feedback computer vision (average score of 2,06), behavior execution (2,0), inter-robot communication (1,85) and low-level device drivers (OS dependent, e.g., USB stack, 1,76) are the most affected software parts.

Table 3 left shows the relevance of causes for faults in the software. The major causes are configuration problems (average score of 2,56), performance leaks (i.e., miss of deadlines, 1,88), memory leaks (1,31) and access violations (1,19). Mutual exclusions (score of 0,64) and overflows (0,57) play only a minor role as causes.

Of particular interest is the relation of causes to faults within the software. About 30 to 70 % of the groups report a relation between configuration problems and failures in different parts of the software. Performance leaks are named as cause for faults in object tracking (60 %), computer vision (54 %) and inter-robot communication (50 %).

Table 3. Ranking the relevance of causes for robot software faults (left) and algorithms used affected by faults (right)

Causes	Ranking (0 .. not relevant. 5 denotes much relevant.)						
	0	1	2	3	4	5	avg.
	%						
Configuration	12,5	12,5	18,8	37,5	0	18,8	2,56
Performance Leaks	25,0	18,8	31,5	0	18,8	6,31	1,88
Memory Leaks	31,3	25,0	25,0	18,8	0	0	1,31
Access Violation	18,8	50,0	2,0	6,3	0	0	1,19
Race Conditions	60,0	26,7	6,7	0	6,7	0	0,67
Mutual Exclusions	78,8	0	14,3 0	0	0	7,1	0,64
Overflows	64,3	21,4	7,1	7,1	0	0	0,57

Algorithm	Ranking (0 .. not affected. 5 .. much affected)						
	0	1	2	3	4	5	avg.
	%						
Decision Making - State Machine	0	33,3	33,3	0	0	33,3	2,67
Object Recognition 2D	0	28,6	21,4	28,6	14,3	7,1	2,50
Feature Extraction	7,7	23,1	30,8	23,1	7,7	7,7	2,23
Mapping 3D	0	60,0	0	20,0	20,0	0	2,00
Mapping 2D	0	69,2	0	15,4	0	15,4	1,92
Classification	0	55,6	11,1	22,2	11,1	0	1,89
Path Planning	7,7	53,8	23,1	0	7,7	7,7	1,69
Path Execution	0	66,7	8,3	16,7	8,3	0	1,67
Self-Localization - Sample Based	21,4	28,6	21,4	21,4	7,1	0	1,64
Object Recognition 3D	0	25,0	25,0	25,0	0	25,0	2,75
Reasoning/Planning - Logic Based	0	33,3	33,3	0	33,3	0	2,33
Reasoning/Planning - Probability Based	0	50,0	50,0	0	0	0	1,50
On-Line Machine Learning	33,3	0	66,7	0	0	0	1,33
Self-Localization - Classical Filter	33,3	33,3	33,3	0	0	0	1,00
Decision Making - Fuzzy Logic	0	100,0	0	0	0	0	1,00
Knowledge Base	50,0	25,0	25,0	0	0	0	0,75

The feedback on the relevance of symptoms (failures) caused by software faults shows that the most prominent symptom is unpredicted behavior (average score of 3,00) followed by limited functionality (2,69) and crashes (1,94). While most faults of the robot software are classified as repairable (by 40 to 70 % of the groups) problems with OS-related low-level drivers (50,0 %), embedded software in actuators and sensors (46,2 %) and self-localization (37,5 %) are classified mainly as terminal to missions. Moreover, the occurrence of most faults is classified as never and sporadic. Only object tracking (by 30,8 % of the groups), low-level drivers (28,6 %) and computer vision (21,4 %) are classified as regularly.

5.7 Algorithms

Table 3 right shows the feedback on how much various algorithms are affected by faults. Please note that the table is divided into two parts. The algorithms of the upper section are commonly used and we thus got a significant number of replies. The algorithms in the lower section are less common and we thus got only a few replies (2 to 4). Please note that this question regards the algorithm itself rather than its implementation.

The most affected commonly used algorithms are decision making with state machines (average score of 2,67), object recognition in 2D (2,50) and feature extraction (2,00). The upper section of the table clearly shows that these algorithms get higher average score. Even the lowest score of 1,64 for sample-based self-localization is significantly higher than the lower scores for the hardware. These observations lead to the interpretation that algorithms are in general more affected by faults than hardware.

The most relevant cause for faults in algorithms is high computational demands (i.e., missing deadlines). It got an average score of 2,8 out of 5. It is followed by uncertain estimations (2,43), false positives (2,33) and wrong estimations (2,21).

Counting how often groups report causal relations between symptoms (failures) and a particular algorithm sample-based self-localization was named as the most affected algorithm (34 reported relations to some cause) followed by 2D mapping (25) and 2D object recognition (23). Please note that one group may report several causes for a problem of a single algorithm. Configuration problems were most often reported as cause for failures (36 reported relations to some algorithm) followed by wrong estimations (34) and missed computation deadlines (27).

Most of the problems of algorithms were mainly classified as repairable. But problems with decision making using state machines were classified by 50 % of the groups as terminal to the mission. It is followed by plan execution (36,4 %) and 2D mapping (33,3 %). Most of the faults of algorithms occur sporadically. Only 2D object recognition was reported by 45,5 % of the groups as regular problem. Decision making with state machine was reported by 12,5 % of the groups as frequent problem. Please note that classifications of algorithms founded only on 1 to 3 responses were omitted.

5.8 Fault Mitigation Techniques

For this part of the survey we asked about techniques the groups already use to mitigate faults. The questions were divided into techniques that are used off-line prior to the mission or on-line during the mission and into techniques for hardware or software. This section is quite important as it returns information about the state of the art in applied fault mitigation that in turn is close related to dependability.

Table 4 right shows the numbers of groups using particular off-line techniques to mitigate faults in the hardware. The most used technique is testing (used by 76,5 % of the groups) followed by preventive maintaining used by 52,8 % of the groups. Advanced techniques such as finite element method are not used.

Table 4. Groups using off-line techniques to handle hardware faults (left) and on-line techniques to mitigate software faults (right)

Technique	groups use it	
	#	%
Test Process	13	76,5
Preventive Maintenance	9	52,9
Simulation	5	29,4
Special Design Process	4	23,5
Special Implementation Process	4	23,5
Redundant Design	3	17,6
Iterative Design Process	1	5,9
None	1	5,9
Finite Element Method (FEM)	0	0

Technique	groups use it	
	#	%
Process Monitoring	13	76,5
Watchdog	10	58,8
Activity Monitoring	10	58,8
Plausibility Check of Results	7	41,2
Automated Diagnosis	5	29,54
Automated Repair	4	23,5
Automated Reconfiguration	2	11,8
Fault Detection and Isolation (FDI)	1	5,9
Automated Degradation/Adaptation	0	0
None	0	0

Major technique used on-line to deal with hardware faults are watchdogs used by 76,5 % of the groups. It is followed by automated diagnosis used by 29,4 % of the groups. Major techniques used off-line to mitigate software faults are testing (88,2 % of the groups) followed by simulation (76,5 %) and special design processes (47,1 %).

Table 4 left shows the numbers of groups using particular on-line techniques to mitigate faults in the software. It shows that mainly monitoring techniques such process monitoring (by 76,5 % of the groups), watchdogs (58,8 %) and activity monitoring (58,8 %) are used. An interesting aspect is that every group takes measures to deal with these on-line problems suggesting that there is a certain level of awareness to the problems.

Based on additional feedback of the groups using automated diagnosis most groups use some heuristics to check if particular sensors or actuators works correctly and simply reset the device by a command or even a power down/up cycle. One group uses additional sensors to validate results of the self-localization and reinitialize it in case. One group uses an logical framework to reason about undesired states of components or the high-level control and to issue repair actions, e.g., recalibration of an arm [15].

6 Conclusion and Future Work

In this paper we motivated, that in order to maintain a certain level of dependability of autonomous robot systems it is important to identify and to deal with faults of the system. In order to point out which faults of the system are more relevant in terms of their frequency and impact on mission success we adapted a fault taxonomy used for remote controlled robots towards autonomous robots. Moreover, we designed and conduct a survey about the nature of faults in the context of RoboCup. We sent the survey to 68 research groups around the world participating in RoboCup regularly. 17 responses were included in our study. The survey comprised the following parts: information about used hardware and software, hardware faults , software faults, and faults of algorithms as well as used counter-measures. The result of the survey is a database about the nature of faults occurring in autonomous robots. All data are publicly available on the survey website for use by other groups (see `http://www.ist.tugraz.at/rfs`).

We now summarize some of the major observation we can draw from the collected information. Faults in sensors have a similar frequency of occurrence as faults in the robot platforms but their negative impact on the success rate of the mission is much higher. Surprisingly, rather simple causes like connector problems causing hardware failures were reported very often. Therefore, one conclusion is that a high fraction of problems can already be mitigated by better engineering.

The involved groups reported that algorithms are in general more affected by faults than hardware. Basically the awareness of the research groups to the problem of dependability and faults is fortunately quite high. Almost all groups use some techniques to mitigate faults in the robot's hardware and software. Nevertheless, mostly straight forward techniques such as watchdogs or monitoring are used. But the fault properties and the interaction of symptoms and faults ask for advanced fault management.

Failures of algorithms caused by missed deadlines were reported by several of the involved groups. Therefore, any-time or at least predictable algorithms seem to be a promising research direction. Moreover, configuration seems to be a major problem and suggests further research in the direction of configuration management. Furthermore, faults related to the properties of algorithms received high relevance scores asking for more research into the direction of evaluation and validation of algorithms.

The survey provides a first qualitative overview on the nature of faults in the RoboCup context. We are convinced that these observations give information useful for developers and researchers in the area of autonomous robot systems in general. For instance researcher may use the information to concentrate on subsets of fault with high impact on dependability or use discovered symptom-cause relationships to improve their diagnosis models.

In future work we will enhance the survey as for example questions concerning fault mitigation techniques have to be more specific. We plan to conduct the revised survey again in RoboCup and other domains to collect more information to be able to provide significant quantitative results in the future. Moreover, the integration of data from industry is of particular interest. Our vision is that the survey will form a basis for a broader evaluation and classification of faults in robot systems.

Acknowledgment. We like to heartily thank all teams from the RoboCup Middle Size, Robot Rescue and RoboCup@Home leagues who actively took part of this survey. The work has been partly funded by the Austrian Science Fund (FWF) by grant P22690.

References

1. Visser, U., Burkard, H.: RoboCup: 10 Years of Achivements and Future Challenges. AI Magazine 28(2) (2007)
2. Carlson, J., Murphy, R.R.: How UGVs Physically Fail in the Field. IEEE Transactions on Robotics 21, 423–437 (2005)
3. Steinbauer, G., Mörth, M., Wotawa, F.: Real-time diagnosis and repair of faults of robot control software. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 13–23. Springer, Heidelberg (2006)
4. Brandstötter, M., Hofbaur, M., Steinbauer, G., Wotawa, F.: Model-based fault diagnosis and reconfiguration of robot drives. In: IEEE Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA (2007)
5. Tomatis, N., Terrien, G., Piguet, R., Burnier, D., Bouabdallah, S., Arras, K.O., Siegwart, R.: Designing a Secure and Robust Mobile Interacting Robot for the Long Term. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan (2003)
6. Bererton, C., Khosla, P.: An analysis of cooperative repair capabilities in a team of robots. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA), vol. 1, pp. 476–482 (May 2002)
7. Verma, V., Gordon, G., Simmons, R., Thrun, S.: Real-time fault diagnosis. Robotics & Automation Magazine 11(2), 56–66 (2004)
8. Long, M., Murphy, R., Parker, L.L.: Distributed multi-agent diagnosis and recovery from sensor failures. In: Int. Conference on Intelligent Robots and Systems (IROS) (2003)
9. Alami, R., Chatila, R., Fleury, S., Ghallab, M., Ingrand, F.: An architecture for autonomy. Intenational Journal of Robotics Research 17, 315–337 (1998)
10. Volpe, R., Nesnas, I.A.D., Estlin, T., Mutz, D., Petras, R., Das, H.: CLARAty: Coupled Layer Architecture for Robotic Autonomy. Technical report, NASA - JPL (2000)
11. Muscettola, N., Nayak, P.P., Pell, B., Williams, B.C.: Remote Agent: to boldly go where no AI system has gone before. Artificial Intelligence 103(1-2), 5–47 (1998)
12. Lauer, M., Riedmiller, M.: Participating in Autonomous Robot Competitions: Experiences from a Robot Soccer team. In: IJCAI 2009 Workshop on Competitions in Artificial Intelligence and Robotics, Pasadena, CA, USA (2009)
13. Kleiner, A., Dornhege, C.: Real-time localization and elevation mapping within urban search and rescue scenarios: Field Reports. Journal of Field Robotics 24, 723–745 (2007)
14. Wisspeintner, T., van der Zant, T., Iocchi, L., Schiffer, S.: RoboCup@home: Scientific Competition and Benchmarking for Domestic Service Robots. Interaction Studies. Special Issue on Robots in the Wild 10(3), 392–426 (2009)
15. Schiffer, S., Wortmann, A., Lakemeyer, G.: Self-Maintenance for Autonomous Robots controlled by ReadyLog. In: Proceedings of the 7th IARP Workshop on Technical Challenges for Dependable Robots in Human Environments, Toulouse, France, pp. 101–107 (2010)