Improving Dependability of Humanoids

Rainer Bischoff, Somrak Petchartee, Volker Graefe

Bundeswehr University Munich, The Institute of Measurement Science
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
E-Mail: Bischoff@ieee.org, URL: http://www.unibw-muenchen.de/hermes

Abstract
A service robot of anthropomorphic size and shape has been built to study dependable ways of interaction with people and their common living environment. Although the robot is presently used mostly by trained personnel it has also shown robust and safe behavior with novice users and people who are not necessarily interested in robotic matters, e.g. at trade fairs, television studios and at various demonstrations in our institute environment. During the design process we followed certain guidelines in both hardware and software that have proved to lead to a reliable and safe overall system. These guidelines are introduced, and some experiments carried out with the real robot in an office environment are presented. We do not claim that our present system is failsafe and foolproof, but we believe that the strategies we embarked on could lead the way to robots having these characteristics.

1 Introduction
“Dependability” is a system concept that integrates such attributes as reliability, availability, safety, confidentiality, integrity, and maintainability [Laprie 1992]. The goals behind the concept of dependability are the abilities of a system to deliver a service that can justifiably be trusted and to avoid failures that are more frequent or more severe, and outage durations that are longer, than is acceptable to the user(s).

Since our society largely depends on infrastructures that are controlled by embedded information systems, the dependability concept has been widely employed for these kind of systems. Although future service and personal robots are supposed to become an important part in our future society, dependability aspects have been almost constantly neglected by researchers. However, dependability concepts are needed especially for these types of robots because they are intended to operate in unpredictable and unsupervised environments and in close proximity to, or in direct contact with, people who are not necessarily interested in them, or, even worse, who try to harm them by disabling sensors or playing tricks on them.

Dependability has not been a major issue in research institutions so far because it is believed that industrial companies, when they will actually market service or personal robot products, will eventually deal with this question. Researchers in laboratories have always been satisfied if their robots performed well once or twice under specific conditions or at end-of-project demonstrations, which enabled them to write a publication about their “perfectly” performing robot. Unfortunately, these “performances” make people (sponsors, public) believe that most of the robotic community’s problems are already solved, which is certainly not true. On the contrary, much research is still needed to improve considerably not only system reliability and safety concepts, but also design concepts, locomotion and manipulation capabilities, cooperation and communication abilities, reliability, and – probably most importantly – adaptability, learning capabilities and sensing skills.

To advance research in all of the before-mentioned areas we have developed the humanoid experimental robot HERMES. It is built from 25 motor modules with identical mechanical and electrical interfaces, thus yielding a very flexible, extensible and modular design that can be easily modified and maintained (Figure 1). With its omnidirectional undercarriage, body, head, eyes and two arms it has now 22 degrees of freedom and resembles a human in height and shape. Its main exteroceptive sensor modality is stereo vision. Both camera “eyes” may be actively and independently controlled in pan and tilt degrees of freedom. Proprioceptive sensors add to its perceptual abilities. A multimodal human-friendly communication interface built upon natural language and the basic senses – vision, touch and hearing – enables even non-experts to intuitively control the robot.

2 Design Strategies
In our opinion dependability of robots emerges from the following simple design strategies and guidelines:
1. Learning from nature how to design reliable, robust and safe systems.
2. Providing natural and intuitive communication and interaction between the robot and its environment.

Figure 1: Humanoid experimental robot HERMES with an omnidirectional undercarriage, a bendable body, two arms with two-finger grippers and a 6-DOF stereo vision system; size: 1.85 m x 0.70 m x 0.70 m; mass: 250 kg, low center of gravity provides good stability.
3. System reliability depends on ease of maintenance.  
4. Only a nice-looking robot is a reliable robot.  

We believe that future robotic systems could benefit from applying these design strategies and guidelines in addition to general design rules that must be followed by the designer of any robotic system with respect to the application domain. In the sequel these design strategies are explained in greater detail.

**Nature has provably designed reliable, robust and safe systems.** According to the classical approach, robot control is model-based. Numerical models of the kinematics and dynamics of the robot and of the external objects that the robot should interact with, as well as quantitative sensor models, are the basis for controlling the robot's motions. The main advantage of model-based control is that it lends itself to the application of classical control theory and, thus, may be considered a straightforward approach. The weak point of the approach is that it breaks down when there is no accurate quantitative agreement between reality and the models. Differences between models and reality may come about easily; an error in one of the many coefficients that are part of the numerical models suffices. Among the many possible causes for discrepancies are initial calibration errors, aging of components, changes of environmental conditions, such as temperature, humidity, electromagnetic fields or illumination, maintenance work and replacement of components, to mention only a few. Consequently, most robots work only in carefully controlled environments and need frequent maintenance (including repetitive calibration), in addition to a cumbersome and expensive initial calibration.

Organisms, on the other hand, are robust and adapt easily to changes of their own conditions and of the environment. They never need any calibration, and they normally do not know the values of any parameters related to the characteristics of their "sensors" or "actuators". Obviously, they do not suffer from the shortcomings of model-based control which leads us to the assumption that they use something other than quantitative measurements and numerical models for controlling their motions. Perhaps their motion control is based on a holistic assessment of situations for the selection of behaviors to be executed. Perhaps robotics could benefit from following a similar approach.

Following this line of argumentation we strongly believe that sensing in general should be based on the senses that have proved their effectiveness in nature. Therefore, vision – the sensor modality that predominates in nature – is also an eminently useful and practical sensor modality for robots. Also, tactile sensing and hearing may greatly improve a robot's safe operation as shown by nature. Active sensing (laser, radar, sonar) might be a suitable approach in the short run for specific system solutions, but only a more generic approach with low-cost universally applicable (passive) sensor modalities on the robot will lead in the long run to the deployment of service and personal robots in massive numbers.

In addition, passive sensors cannot harm human eyes, ears or tissue, whereas active sensors could be hazardous.

**Natural and intuitive communication and interaction enhances safety.** Any person who might, voluntarily or not, encounter a service robot needs to be able to communicate and interact with the robot in a natural and intuitive way. Therefore, the human communication interface has to be designed in a way that no training would be required for any person who might get in contact with the robot. This can be achieved if the human-robot communication would resemble a dialogue that could as well take place between two humans. If the robot resembles a human, a person can easily derive from his former everyday experience with humans how a specific interaction, e.g., exchanging objects with the robot, might work. Even if the robot does not have humanoid shape, a safe confidence-inspiring interaction could benefit from humanoid characteristics such as smoothness of movements and compliance of the joints or links. In general, unexpected robot movements should be avoided. Instead, gentle human-like motions should be generated to enable operators or uninvolved persons to anticipate the robot's actions.

It would be dangerous, however, to try to anticipate people's movement in order to let the robot operate faster. Since humans might behave in illogical, irrational or unpredictable ways, it is necessary to have the robot move and interact in a way that prevents accidents under all circumstances.

Therefore, it might be useful to additionally visualize the robot's state or subsequent motions in a way that facilitates anticipation, e.g., with help of facial expressions, postures or even indicators that humans are familiar with in everyday situations. Doing so, it should be the goal to exploit the people's own intuition to make the interaction safer.

**System reliability depends on ease of maintenance.** In our opinion the first step to make a complex system safe is to make its components reliable. If the components themselves are failsafe and need little or no maintenance at all, overall system safety is greatly increased. We believe that only a robot that needs little or no maintenance at all and that can be easily repaired (if ever needed) will be accepted as a co-worker, caretaker or companion. This requires, among other things, enclosed and maintenance-free subsystems such as the modules used to build the robot's joints.

**Only a tidy-looking robot can be a reliable robot.** It is a matter of personal experience that, especially in research environments, robots often fail because of broken cables and unreliable connections. Such robots often look very cluttered with cables criss-crossing each other, and circuitry and connectors hidden under bundles of wires. This not only makes visual inspection difficult, but it may also be taken as an indication that the persons who built and maintain the robot have only placed little emphasis on the initial design stage. Although software is not visible, the observer wonders whether the structure of the robot's software might resemble the layout of the robot's wiring.

If, on the other hand, the designer has made an effort to have the robot look tidy, it may be assumed that he also has taken care to do other things right, such as reliably connecting the different sensors, actuators and peripherals, finding proper ways to route all cables - and maybe even structure the software in a systematic way. Of course, a good design involves more than aesthetic aspects. Industrial designers,
e.g., consider all aspects from ergonomics over construction to deployment. Nevertheless, it should be mentioned that only a few research institutes really try to consider these aspects in a holistic fashion to provide a truly robust system. If robotics researchers placed more emphasis on the appearance of their robots, their robots might become easier to maintain, more robust and, thus, more dependable as a side effect.

3 Dependable Robot Hardware

In designing our humanoid experimental robot we placed great emphasis on modularity and extensibility [Bischoff 1997]. All drives are realized as modules with compatible mechanical and electrical interfaces; each drive module consists of two cubes rotating relative to each other and containing a motor, a Harmonic Drive gear, power electronics, sensors, a micro-controller, and a communication interface. All modules are connected via CAN bus with the main computer.

HERMES runs on 4 wheels, arranged on the centers of the sides of its base. The front and rear wheels are driven and actively steered, the lateral wheels are passive. The manipulator system consists of two articulated arms with 6 degrees of freedom each on a body that can bend forward (130°) and backward (-90°). The work space extends up to 120 cm in front of the robot. The heavy base guarantees that the robot will not lose its balance even when the body and the arms are fully extended to the front. Currently each arm is equipped with a two-finger gripper that is sufficient for basic manipulation experiments.

Main sensors are two video cameras mounted on independent pan/tilt drive units in addition to the pan/tilt unit that controls the common “head” platform. The cameras can be moved with accelerations and velocities comparable to those of the human eye.

A radio Ethernet interface allows to control the robot remotely. A wireless keyboard can be used to teleoperate the robot up to distances of 7 m. Separate batteries for the motors and the information processing system allow a continuous operation of the robot for several hours without recharging.

A hierarchical multi-processor system is used for information processing and robot control. The control and monitoring of the individual drive modules is performed by the sensors and controllers embedded in each module. The robot’s “brain” is a network of digital signal processors (DSP, TMS 320C40) embedded in a rugged PC. Sensor data processing (including vision), situation recognition, behavior selection and high-level motion control are performed by the DSPs, while the PC provides data storage and the human interface.

A number of special hardware measures have been taken to enhance reliability and operating safety. They are described in the sequel.

Modular and standardized computer hardware. Ease of maintenance and repair is certainly one of the most prominent features of HERMES, since the robot consists of 25 functionally similar drive modules with almost identical mechanical and electrical interfaces. If any of these modules should ever fail, it could be easily replaced with a new readily available off-the-shelf module. Same holds true for the robot’s brain: each DSP board and the single slot CPU can be easily replaced from stock. A rugged PC with special shielding and ventilation keeps the processors’ temperatures down and reduces electromagnetic noise to a minimum.

Cables and connectors. Within HERMES, all signal and power line connectors are secured with screws or fixtures to their respective housing. All connectors are strain-relieved to eliminate the risk of loose or broken cables. Electromagnetic shielding of the cabling has been a major concern to diminish the effect of the many sources of electromagnetic fields within the robot.

Power circuitry and emergency stopping. Standard safety regulations for industrial robots require that all consumer loads are disconnected from the power in case of an emergency and that all drives are actively braked, e.g., if the bumpers are touched. In this case a human operator is needed to reset the robot. Any kind of intelligent assessment of the prevailing “emergency” situation by the robot is not allowed. However, in normal living environments the robot might need to touch things or cannot prevent it if it wants to continue its given task. Should it not have the ability to intelligently assess the situation? For instance, maybe it would suffice during simple maneuvers such as turning around a corner just to back up a little bit or to change the curvature in order to prevent any damage to the walls. Another scenario could require to reset the robot’s modules into a compliant mode where all joints can be moved manually with ease to prevent further injury to a human instead of actively braking all drives. We believe that future robots need to have more intelligent safety concepts than the existing ones to be able to work with, or in close proximity to, humans. It will be simply not safe enough to just follow the existing safety regulations for industrial manipulators or automated guided vehicles.

Therefore, our safety concept allows active utilization of the bumpers to enable tactile sensing and to complement missing visual information. Program failures could be detected by implementing so called “watch dog” timers on different levels, e.g., in the robot’s microcontrollers, the slot CPU and DSPs. Any watch dog timer running out would cause the robot to stop via electronic emergency switches. So far, these watch dog timers have been implemented on DSP and PC level only.

HERMES possesses two standard emergency buttons. One may be activated by pressing a clearly visible red-yellow button on the robot’s cargo area, another one may be activated via a wireless emergency switch carried by a human operator. They are connected in series and only interrupt the power circuitry for the motors; the information processing system keeps running as long as the robot is switched on. No time is wasted in case of an emergency to “reboot” the robot.

On a lower level, current sensors in each module check if the motor current is too high. In this case the power line
for that module will be interrupted to prevent further damage to the electronic components, and a break is activated to prevent grasped objects from falling down.

**Artificial skin.** A modular approach has also been taken in the design of an artificial skin for the robot. This “skin” is based on conductive foam that serves two purposes: one, it damps accidental and unwanted impacts between the robot and humans or environmental objects, and two, it allows to identify the contact locations of, and the forces exerted by, the touched objects. Contact points and forces are measured via a dense grid of electrodes underneath the foam. Pressing the foam results in a higher conductivity of the material (lower resistance, respectively). The resistance between each electrode and a ground plane is continuously measured (sample rate 50 Hz) and evaluated by dedicated microcontrollers.

In case of touch events these microcontrollers first send messages to higher hierarchical computing levels that decide about appropriate reactions based on the robot’s current situation. If for any reason these higher levels do not immediately respond to the message, the microcontroller will directly stop the associated motor module(s).

A bumper consisting of 12 identical modules of the artificial skin surrounding the robot’s undercarriage (at a height of 30 - 330 mm, each section 200 mm wide) has already been realized. Furthermore, two new two-finger grippers that are completely covered by this conductive foam have been developed and are currently being integrated (Figure 2). In future it is planned to cover the whole robot structure with this kind of tactile sensing elements that can be easily replaced. Ideally, these elements will be directly connected to the individual motor modules and connected via a safe bus system to the central information processing unit. Presently, they are connected via a high-speed serial communication bus (CAN).

![Figure 2: Developed tactile sensing elements: Top Left: one of twelve identical bumper modules covering the undercarriage at a height between 30 and 330 mm; Bottom Left: side view of the bumper module: the electronic circuits are integrated in, and thus, protected by an epoxy layer; modules are connected via a serial bus through a cable duct; Right: Finger module: printed circuit boards (PCBs) are glued with epoxy to an aluminum core; a dense grid of electrodes covers the surface on all four sides and the tip of the finger; PCBs are covered by 3 mm thick conductive foam (not shown).](image)

Another (or a complementing) solution could be to employ either slip clutches in the joints of manipulators or to implement intelligent control algorithms that continuously predict and verify force and torque on all joints. Prerequisite for the latter safety concept would be a lightweight manipulator that allows position, velocity and torque control with minimal control loop cycle times (<1 ms). This is currently unfeasible with off-the-shelf industrial products. Therefore, it is definitely advisable for the time being to use some kind of high-resolution tactile sensors to reliably detect (un)wanted contacts of the robot and its environment.

**Extended dynamic range of CCD cameras.** To increase the robustness of the image processing the robot’s two CCD cameras have been modified to allow their integration time, and thus their sensitivity, to be controlled by the vision system. This enables the robot to reliably detect objects even under uncontrolled and changing lighting conditions by maintaining a high contrast around tracked features or keeping an average grey level within a region of interest. Automatic gain control which is usually based on an average grey level within the entire image does not yield satisfactory results because it cannot cope with the high differences in brightness of natural scenes which leads to over-/underexposure of regions of interest or tracked objects.

### 4 Dependable Robot Software and System Architecture

Seamless integration of many – partly redundant – degrees of freedom and various sensor modalities in a complex robot calls for a unifying approach. We have developed a system architecture that allows integration of multiple sensor modalities and numerous actuators, as well as knowledge bases and a human-friendly interface. In its core, the system is behavior-based, which is now generally accepted as an efficient basis for autonomous robots [Arkin 1998]. However, to be able to select behaviors intelligently and to pursue long-term goals in addition to purely reactive behaviors, we have introduced a situation-oriented deliberative component that is responsible for situation assessment and behavior selection.

#### 4.1 System Overview

Figure 3 shows the essence of the situation-oriented behavior-based robot architecture as we have implemented it. The situation module (situation assessment & behavior selection) acts as the core of the whole system and is interfaced via “skills” in a bidirectional way with all other hardware components – sensors, actuators, knowledge base storage and MMI (man-machine, machine-machine interface) peripherals.

These skills have direct access to the hardware components and, thus, actually realize behavior primitives. They obtain certain information, e.g., sensor readings, generate specific outputs, e.g., arm movements or speech, or plan a route based on map knowledge. Skills report to the situation module via events and messages on a cyclic or interruptive basis to enable a continuous and timely situation update and error handling.
The situation module fuses via skills data and information from all system components to make situation assessment and behavior selection possible. Moreover, it provides general system management (cognitive skills). Therefore, it is responsible for planning an appropriate behavior sequence to reach a given goal, i.e., it has to coordinate and initialize the in-built skills. By activating and deactivating skills, a management process within the situation module realizes the situation-dependent concatenation of elementary skills that lead to complex and elaborate robot behavior.

In general, most skills involve the entire information processing system. However, at a gross level, they can be classified into five categories besides the cognitive skills: Motor skills are simple movements of the robot’s actuators. They can be arbitrarily combined to yield a basis for more complex control commands. Encapsulating the access to groups of actuators, that form robot parts, such as undercarriage, arms, body and head, leads to a simple interface structure, and allows an easy generation of pre-programmed motion patterns. Sensor skills encapsulate the access to one or more sensors, and provide the situation module with proprioceptive or exteroceptive data. Sensorimotor skills combine both sensor and motor skills to yield sensor-guided robot motions, e.g., vision-guided or tactile and force/torque-guided motion skills. Communicative skills pre-process user input and generate a valuable feedback for the user according to the current situation and the given application scenario. The system’s knowledge bases are organized and accessed via data processing skills. They return specific information upon request and add newly gained knowledge (e.g., map attributes) to the robot’s data bases, or provide means of more complex data processing, e.g., path planning. For a more profound theoretical discussion of our system architecture which bases upon the concepts of situation, behavior and skill see [Bischoff, Graefe 1999].

4.2 Implementation

A robot operating system has been developed that allows sending and receiving messages via different channels among the different processors and microcontrollers. All tasks and threads run asynchronously, but can be synchronized via messages or events.

Overall control is realized as a finite state automation that does not allow unsafe system states. It is capable of responding to prioritized interrupts and messages. After powering up the robot finds itself in the state “Waiting for next mission description”. A mission description is provided as a text file that may be either loaded from a disk, received via e-mail, entered via keyboard, or result from a spoken dialogue. It consists of an arbitrary number of single commands or embedded mission descriptions that let the robot perform a required task. All commands are written or spoken, respectively, in natural language and passed to a parser and an interpreter. If a command cannot be understood, is under-specified or ambiguous, the situation module tries to complement missing information from its situated knowledge or asks the user via its communicative skills to provide it.

Motion skills are mostly implemented at the microcontroller level within the actuator modules. High-level motor skills, such as coordinated smooth arm movements are realized via a dedicated DSP interfaced to the microcontrollers via a CAN bus. Sensor skills are implemented on those DSPs that have direct access to digitized sensor data, especially digitized images.

4.3 Special software measures for enhancing safety and operating robustness

Object-oriented image processing. One apparent difficulty in implementing vision as a sensor modality for robots is the huge amount of data generated by a video camera: about 10 million pixels per second, depending on the video system used. Nevertheless, it has been shown that modest computational resources are sufficient for realizing real-time vision systems if a suitable system architecture is implemented [Graefe 1989].

As a key idea for the design of efficient robot vision systems the concept of object-oriented vision was proposed. It is based on the observation that both the knowledge representation and the data fusion processes in a vision system may be structured according to the visible and relevant external objects in the environment of the robot. For each object that is relevant for the operation of the robot at a particular moment the system has one separate “object process”. An object process receives image data from the video section (cameras, digitizers, video bus etc.) and generates and updates continuously a description of its assigned physical object. This description emerges from a hierarchically structured data fusion process which begins with the extraction of elementary features, such as edges, corners and textures, from the relevant image parts and ends with matching a 2-D model to the group of features, thus identifying the object.

Recognition of relevant objects is crucial for the robot’s operation. The decision what objects have to be detected and tracked is made by the situation module. It also decides that the robot has to move slower, if, e.g., some features are
Learning by doing. Two forms of learning are currently being investigated. They both help the robot to learn from scratch by actually doing a useful task: One, to let the robot automatically acquire or improve skills, e.g., grasping of objects, without quantitatively correct models of its manipulation or visual system (autonomous learning). Two, to have the robot generate, or extend, an attributed topological map of the environment over time in cooperation with human teachers (cooperative learning).

The general idea to solve the first learning problem is simple. While the robot watches its end effector with its cameras, like a playing infant watches his hands with his eyes, it sends more or less arbitrary control commands to its motors. By observing the resulting changes in the camera images it “learns” the relationships between such changes in the images and the control commands that caused them. After having executed a number of test motions the robot is able to move its end effector to any position and orientation in the images that is physically reachable. If, in addition to the end effector, an object is visible in the images, the end effector can be brought to the object in both images and, thus, in the real world.

Based on this concept a robot can localize and grasp objects without any knowledge of its kinematics or its camera parameters. In contrast to other approaches with similar goals, but based on neural nets, no training is needed before the manipulation is started [Graefe 1995].

The general idea to solve the second learning problem is to let the robot behave like a new worker in an office with the ability to explore, e.g., a network of corridors, and to ask people for reference names of specific points of interest, or to let people explain how to get to those points of interest. The geometric information is provided by the robot’s odometry, and relevant location names are provided by the people who want the robot to know a place under a specific name. In this way the robot learns quickly from scratch, how to deliver personal service, especially: how do (specific) persons call places; what are the most important places and how to get there; where are objects of personal and general interest located; how should specific objects be grasped? The ability to link, e.g., person names to environmental features, requires several databases and links between them in order to obtain the wanted information, e.g., whose office is located where, what objects belong to specific persons and where to find them.

Many types of dialogues exist to cooperatively teach the robot new knowledge and to build a common reference frame for subsequent execution of service tasks. For instance, the robot’s lexical and syntactical knowledge bases can easily be extended, firstly, by directly editing them (since they are text files), and secondly, by a dialogue between the robot and a person, thus adding new words, argument classes and prototypes. Also, new macro commands can be learned during run time.

To teach the robot names of persons, objects and places that are not yet in the database (and, thus, cannot be understood by the speech recognition system), a spelling context has been defined that mainly consists of the international spelling alphabet. It has been optimized for ease of use by humans in noisy environments, such as aircraft, and has proved its effectiveness for our applications as well.

Speaker-independent voice recognition. The robot understands natural continuous speech independently of the speaker, and can, therefore, be commanded in principle by any non-dumb human. This is a very important feature, not only because it allows anybody to communicate with the robot without needing any training with the system, but more importantly because the robot may be stopped by anybody via voice in case of emergency. Speaker-independence is achieved by providing grammar files and vocabulary lists that contain only those words and provide only those command structures that can actually be understood by the robot.

In the current implementation HERMES understands ca. 60 different command structures and 350 words in each of the available three languages English, German and French.

Robust dialogues for dependable interaction. Most parts of robot-human dialogues are situated and built around robot-environment or robot-human interactions, a fact that has been exploited to enhance the reliability and speed of the recognition process by using so-called contexts. They contain only those grammatical rules and word lists that are needed for a particular situation. However, at any stage in the dialogue a number of words and sentences not related to the current context are available to the user, too. These words are needed to “reset” or bootstrap a dialogue, to trigger the robot’s emergency stop and to make the robot execute a few other important commands at any time.

It is important to note that the robot is always in charge of the current action and controls the flow of information towards the user. If the robot is asked by a user to execute a service task it will follow a specific “program” consisting of concatenated and combined skills thereby tightly coupling acting, sensing and speech acts in a predefined way. If something goes wrong, i.e., some parameters exceed their bounds, the current command will be canceled by the robot. Canceling a command involves returning into a safe state which again might involve communication and interaction with the user.

Obviously, there are some limitations in our current implementation. One limitation is that not all utterances are allowed or can be understood at any moment. The concept of contexts with limited grammar and vocabulary does not allow for a multitude of different utterances for the same topic. General speech recognition is not sufficiently advanced, and compromises have to be accepted in order to enhance the recognition in noisy environments. Furthermore, in our implementation it is currently not possible to track a speaker’s face, gestures or posture. This would definitely increase the versatility and robustness of human-robot communication.

Learning by doing. Two forms of learning are currently being investigated. They both help the robot to learn from scratch by actually doing a useful task: One, to let the robot...
although its usage is not as intuitive and natural as individual spelling alphabets or as a more powerful speech recognition engine would be.

5 Experiments and Results

We applied the above laid-out design strategies and implemented them in our humanoid service robot HERMES. Two of the peculiarities of our robot are certainly its anthropomorphic size and shape and its modular design. We have experienced that its anthropomorphic shape encourages people to interact with it in a natural way. Besides its appearance, HERMES possesses several other promising features inside and outside that makes it intrinsically more reliable and safer than other robots.

A number of experiments have been carried out in the meantime that prove the suitability of our approach. The robot has been presented at trade fairs, in television studios and at various demonstrations in our institute environment. Due to limited space only a few of the experiments can be described here. The reader may refer to the literature, ([Bischoff, Graefe 1998], [Bischoff 2000]), especially for image processing details, or check the authors’ web pages for latest results.

One of the promising results is that our calibration-free approach seems to pay off, because we experienced drifting of system parameters due to temperature changes or simply wear of parts or aging. These drifts could have produced severe problems, e.g. during object manipulation, if the employed methods relied on exact kinematic modeling and calibration. Since our navigation and manipulation algorithms only rely on qualitatively (not quantitatively) accurate information, the performance of HERMES is not affected by such drifts.

Tactile sensing also greatly improve the system’s dependability. Figure 4 shows an example of the tactile bumper sensors’ response in case of an accident. In this simple contact situation HERMES tries to continue to deliver its service, e.g., to transport an object, and does not wait until a human has solved the problem. In such a simple case the robot would drive backwards, modify the steering angle and try again. More complex contact situations (2 or more contact locations) still require, for safety reasons, the help of a human.

The dialogue depicted in Figure 5 may serve as an example how robots and people in general could build a common, and thus dependable, reference frame in their shared working environment. Whenever a command is incomplete (missing command arguments) or ambiguous (too many arguments or imprecise description), a specific dialogue is initiated to resolve the problem. It is important to note that it is always the robot (except in emergency) who is in charge of the current action and controls the flow of information towards the user.

Autonomously or through dialogues with people, the robot is able to build an attributed topological map of its environment (Figure 6). Since HERMES is using only vision for navigation purposes it is limited by its relatively poor perception (when compared to humans). Nevertheless, the situation-oriented and skill-based system architecture in addition to the camera’s active exposure time control enables a navigation performance that is more than adequate for our office building environment. Combined visual and tactile sensing is only in its early stages. We expect the robot to perform even more dependably when these senses are fully integrated and combined.

6 Summary and Conclusions

A service robot of anthropomorphic size and shape has been built to study dependable ways of a robot’s interaction with people and their common living environment. Although the robot is presently used mostly by trained personnel, it has also shown robust and safe behavior with novice users and people who are not necessarily interested in robotic matters, e.g., at trade fairs, television studios and at various demonstrations in our institute environment. The robot is basically constructed from readily available motor modules with standardized and viable mechanical and electrical interfaces. The resulting modular robot structure makes the robot easy to maintain, which is essential for system dependability from a hardware point of view. System dependability from a software point of view is ensured by a simple but powerful skill-based system architecture that integrates visual, tactile and auditory sensing and various motor skills that do not rely on quantitatively exact models or accurate calibration.

The robustness of the robot’s camera eyes with respect to varying lighting conditions is greatly enhanced by actively controlling the integration time of the CCD sensor elements within an object-oriented software framework, and, thus, allowing safe navigation and manipulation even under uncontrolled and sometimes difficult lighting conditions. Recent efforts include the development of an artificial touch-sensitive skin that can be easily attached to any motor module or outer structure element, such as the undercarriage, grippers or arms. The robot understands spoken natural language speaker-independently, and can, therefore, be commanded in principle by any non-dumb human.

Admittedly, today’s robots (including HERMES) still have limited sensing abilities. Also, the perception quality is not high enough to cope with all kinds of real-world situations.
Cooperative Learning Experiment
In this experiment HERMES and a person are standing in the room commonly referred to as “kitchen”. HERMES has been recently introduced to this and some other rooms and only knows a few people so far. It is in a state of waiting for a new instruction within the (default) service context.

Person: “Hello!”
HERMES has to find out who is requesting a service since he is not yet able to recognize people or faces.
HERMES: “Hello. My name is HERMES. What is your name, please?”
Person: “My name is Rainer.”
HERMES cannot find the person’s name in its database, and therefore, switches to the spelling context.
HERMES: “I am sorry, but I do not know you. Please spell your name!”
Person: “Romeo Alpha India November Echo Romeo.”
HERMES switches to the confirmation context.
HERMES: “Your name is Rainer. Is this correct?”
Person: “Yes!”
HERMES extends its database of persons and switches back to the service context.
HERMES: “Hello Rainer. Pleased to meet you. What can I do for you?”
Person: “Take this glass. Bring it into my office.”
These sentences invoke a number of different skills. First HERMES takes over the glass the person currently holds in his hand. Then, the robot resolves the pronouns “it” and “my” from its situated knowledge. However, the robot does not yet know where the person’s office is located. HERMES has to ask the user to retrieve the missing information.
HERMES: “Please describe the way to your office or give me the room number.”
HERMES switches to the navigation context.
Person: “The room number is 24551”
HERMES switches back to the service context, and starts searching for the room. Alternatively, the person could have given instructions on how to reach the room, such as “Leave the kitchen through the door in front of you. My office is located at the second door to the left!” After having actually found the room, HERMES extends its database of known locations, and marks the room as one of Rainer’s offices.

Figure 5: Excerpt from a dialogue between a human and HERMES to transport an object to another room. In its course, HERMES learns more about its environment and stores this knowledge in several databases for later reference (e.g., the attributed topological map shown in Figure 6). It should be noted how often contexts are switched, depending on the robot’s expectations. This improves the speech recognition considerably.

Anyone who wants to trick the robot can do it, and the robot will fail. The slow progress in this area is certainly due to the complexity of the problem, but also because researchers all over the world are wasting time with building and maintaining their own robotic research platforms and reinventing algorithms again and again. Establishing hardware and safety standards (comparable to PCs and industrial robots), and providing software libraries for already solved perception problems would definitely accelerate research and make the robots more dependable even in the short run.

7 Literature


