

Natural Communication and Interaction with Humanoid Robots

Rainer Bischoff[†] and Tamhant Jain[‡]

[†] Bundeswehr University Munich
The Institute of Measurement Science
85577 Neubiberg, Germany

Phone: +49-89-6004-3587; Fax: -3075

E-Mail: Rainer.Bischoff@unibw-muenchen.de
<http://www.unibw-muenchen.de/campus/LRT6>

[‡] Indian Institute of Technology
Engineering Department

Kanpur, P.I.N. 208016, U.P., India

Phone: +91-512-597-313; Fax: -995

E-Mail: jjain@iitk.ac.in
<http://www.iitk.ernet.in>

Abstract

In the personal or service robotics domain a very close interaction between humans and robots is crucial. To advance research in this field we have designed and built the humanoid robot HERMES. It is shaped according to an anthropomorphic model and is equipped with two arms, a bendable body, a pan-tilt head with two video cameras and an omnidirectional wheelbase. It combines visual, kinesthetic and tactile sensing for enabling a natural communication and interaction with humans. HERMES is capable of speaker-independent speech recognition and speech output. A special behavior-based system architecture is employed to integrate these key technologies. It is based on an understanding of situations for the selection of the behaviors to be executed. Implementing this architecture allows almost natural human-like communication and interaction and makes the robot appear intelligent.

1 Introduction

Although the vast majority of robots today are used in factories, technological advances are enabling specialized robots to automate many tasks in non-manufacturing industries such as agriculture, construction, health care, retailing and other services. These so-called “field and service robots” aim at the fast growing service sector and promise to be a key product for the next decades [Schraft, Schmierer 1998]. All robots working in these domains have been designed to provide special solutions for specific problems. They rely on detailed teaching and programming and carefully prepared environments. It is costly to maintain them and it is difficult to adapt their programming to even slightly changed environmental conditions or modified tasks.

Many humanoid type robots are currently being developed to overcome these limitations and to advance research in such demanding fields as design and safety measures, loco-

motion and manipulation, cooperation and communication, reliability, and – still probably most importantly – adaptability, learning, and perception ([Brooks, Stein 1993], [Konno et al. 1997], [Bergener et al. 1997], [Yamaguchi, Takanishi 1997], [Honda 1997]). We have developed the humanoid robot *HERMES* based on our previously gained experiences [Bischoff, Graefe 1998] to be able to tackle most of these fields and to combine methods and to demonstrate results under a coherent framework. *HERMES* is an excellent research basis for sensor-guided mobile manipulation, environmental learning and effective human-robot communication and cooperation [Bischoff, Graefe 1999].

One of our demonstration scenarios includes a delivery and guidance service in an office-type building with extended networks of corridors. No prior information should be available to the robot before its actual deployment, except that it has the abilities to navigate successfully in corridors and junctions and to detect work stations, such as tables, shelves, mail boxes etc. In order to do useful services for the benefit of its human co-workers the robot should be able to learn more about its working environment. Therefore, the robot needs to be equipped with communication and human-robot-interaction skills besides various other sensorimotor skills. The given task is similar to the introduction of a new colleague into his new working environment. The new colleague will certainly have certain basic abilities, e.g., to “navigate” successfully the network of corridors, but most importantly, he will be able to learn about the environment from reading information signs, observations and asking people.

In this paper we report about our efforts to set up a natural language speech interface that enables *HERMES* to communicate effectively with people in an almost natural way. In the sequel (section 2) we review *HERMES*’ situation-oriented control architecture that is key to the successful implementation of natural communication skills (section 3). Phys-

ical interaction with people is based on several sensorimotor skills which are presented in section 4. The overall system performance may be assessed based on real-world experiments which are described in section 5. Finally, section 6 provides conclusions.

2 The Humanoid Service Robot *HERMES*

2.1 Design and Realization of *HERMES*

In designing our humanoid experimental robot we placed great emphasis on modularity and extensibility [Bischoff 1998]. 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-transmission combination, power electronics, sensors, a micro-controller, and a communication interface. A standardized CAN bus connects all drive modules 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 (Figure 1).

Main sensors are two video cameras on a pan-tilt platform. Numerous proprioceptors such as angle encoders, current converters and temperature sensors are integrated in the motor modules, additional sensors may be connected via available interfaces. 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.

2.2 Behavior-Based Control of a Humanoid Robot

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.

System overview. Figure 2 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 peripherals (man-machine and 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 wheelbase, 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

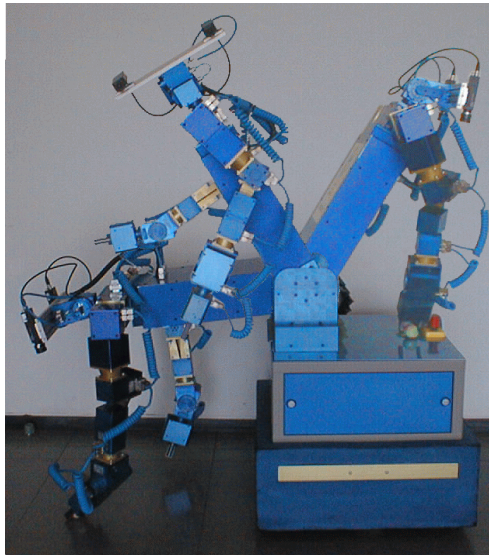


Figure 1: Motion sequence to illustrate the enlarged work space gained by a bendable body. The two arms have 6 degrees of freedom each.

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 on our system architecture which bases upon the concepts of situation, behavior and skill see [Graefe, Bischoff 1997] and [Bischoff, Graefe 1999].

Implementation. A hierarchical multi-processor system is used for information processing and robot control. Monitoring and control of the individual drive modules are performed by the sensors and controllers embedded in each module. The main computer is a network of digital signal processors (DSP, TMS 320C40) embedded in a standard industrial 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 (Figure 3).

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. The two cameras are synchronized via hardware. Image capture may be simultaneous, e.g., when stereo vision is to be performed, or it may run asynchronously, e.g., to provide two different image capture rates for two independent image processing tasks.

Overall control is realized as a finite state machine capable of responding to prioritized interrupts and messages. After powering up the robot finds itself in the state "Waiting for next mission

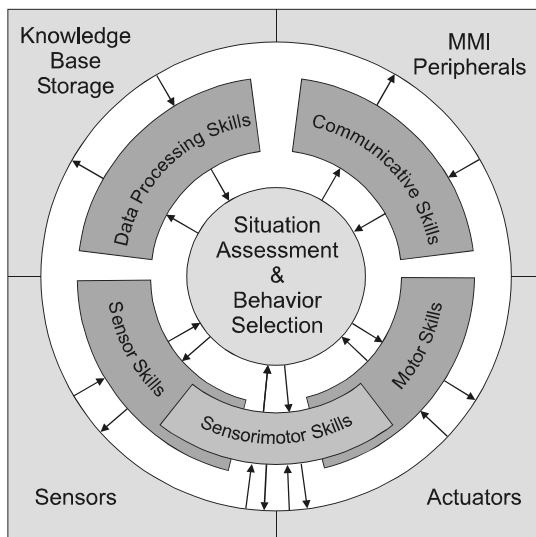


Figure 2: System architecture of a personal robot based on the concepts of situation, behavior and skills.

description". A single command line or mission description with multiple commands are either entered directly via keyboard or spoken into the robot's microphones or provided as a text file. A text file may be either loaded from a disk or received via e-mail. 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 in natural language and passed to a command 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 (details are given in the next section).

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 by 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.

3 Human-Robot Communication

Since natural language is the easiest and most natural mode of communication for a human it is desirable to integrate speech recognition and output into most personal and service robots. Language can be used to instruct the robot with higher-level goals or to intervene certain behaviors and modify their execution. However,

to be accepted as cooperative partners, robots must not only have the ability to understand perfectly clear and complete commands, but they must also resolve ambiguities and complement missing information that is inherent in human conversation. In doing so, an intelligent robot should pursue two approaches: One, it should use the current situation as a relevant context, and two, it may evoke additional information from the human through a dialogue.

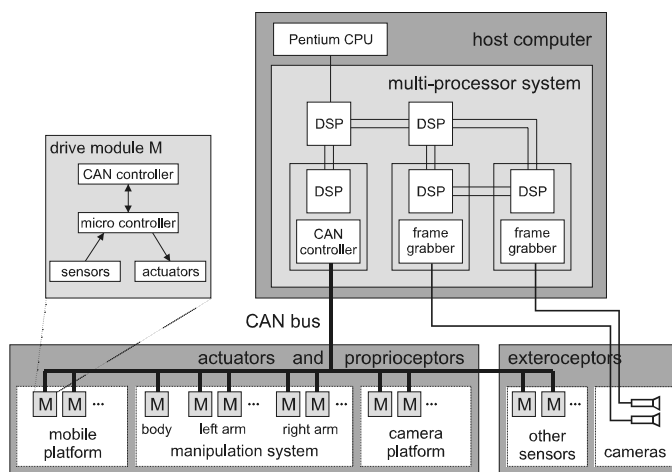


Figure 3: Modular and adaptable hardware architecture for information processing and robot control.

3.1 Communicative Skills

The communicative skills of the robots are mostly based on natural language. Natural language is used both to instruct the robot and to generate easy-to-understand messages for the user. Commands may be input via voice (via a wireless microphone) or keyboard (directly via a wireless keyboard or indirectly via e-mail messages that may contain multiple commands). The robot displays its messages either on a screen or generates speech from text.

To enable natural language processing with limited computational resources, a simple grammar has been designed. It is able to cover all the commands, statements and questions that might be given to the robot. Command sentences have a simple structure allowing them to be classified after the first word, thus facilitating the interpretation of the following words. Examples for command sentences are object and action-oriented instructions such as "Go to the kitchen!", or "Grasp the small ball!". Directive instructions such as "Turn around!" or more complex commands like "Turn left at the next intersection!" are supported as well. Intervening commands that do not contain a command verb are partly supported, e.g., "faster" (instead of "move faster"). In this case these adverbs are treated like single command words. Questions that start with specific key words are allowed as well, e.g., "What", "Where" and "How". Only a few questions can be answered by the robot so far, e.g., "What is your status?", "Where are you?", "How do I get to the kitchen?" etc. The fixed syntax obviously does not allow an arbitrary reordering of parts of the sentences, e.g., "Take the glass, the big one" or "The glass over there, please take it". Those sentences are nevertheless possible, if they had been defined in advance (e.g., using the macro language described below).

3.2 Command Interpreter

A command interpreter handles all user input. It consists of a parser, a lexical analysis, a syntactical analysis and a semantical analysis (Figure 4). The parser is fed by a text string provided by the speech recognition module or a keyboard. It separates the character string into a sequence of words and numbers using space, tabulator and punctuation characters as delimiters. This list is given to the lexical analysis where each word is looked up in a dictionary to obtain its type. Possible types are command verbs (e.g., go, take, place), locations (e.g., office, kitchen, workshop), prepositions (e.g., to, on, onto, in, into), objects (e.g., ball, table, pen) and fill words (e.g., please), just to name a few.

Character strings enclosed in quotation marks are treated as

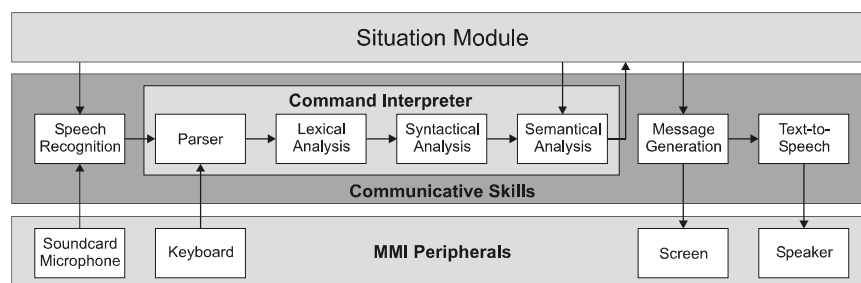


Figure 4: Visualization of the data flow between the peripherals of the man-machine interface, the communicative skills and the situation module. The robot's current situation influences the speech recognition and the semantical analysis of the command interpreter.

one part of a sentence of type "text string". The following syntactical analysis tries to identify the structure of the sentence by comparing the list of types with a list of prototype command sentences that includes all the commands the robot is able to understand. If the comparison is successful the semantical analysis will eventually provide missing words or place holders (such as "it") from the robot's situated knowledge in order to make the command complete.

Language data base. The robot's language data base is fed by two text files: one provides the vocabulary, the other the correct command syntax. The correct syntax is defined by prototype commands that consist of a command word and an arbitrary number of arguments:

COMMAND_WORD argument1 argument2 ... argument_n

The command word is in the imperative form of the verb, e.g., GO, TURN, TAKE, etc. Each argument is a place holder for a specific group of words. Grouping is determined by grammatical or semantical necessities, e.g., the group [preposition] contains all prepositions (to, onto, into,...), the group [object] contains the known objects (ball, pen, table, ...), [location] contains valid places to go to (workshop, office, kitchen, ...), [numbers] contains all valid numbers etc.

A simple command prototype could be 'GO [location]'. To add more flexibility to the command syntax, multiple command categories have been defined. Mandatory arguments are put into square brackets '[]', optional arguments are preceded by a minus sign '-', multiple arguments of the same type are indicated by round brackets '()'. Mandatory extensions of the command words are given directly with no extra character.

For the command word "go" the following prototypes have been defined (among others):

```

GO ([location])
GO -[preposition] [location]
GO -[preposition] [location] VIA ([location])
    
```

These prototypes enable the interpretation of the following commands:

- 1) "go workshop, office, kitchen"
- 2) "go to the kitchen"
- 3) "go to the kitchen via workshop and office"

The first command is interpreted in such a way that the robot would have to go to the locations workshop, office and kitchen (in that order). The ability to

use a preposition in the second instruction lets it resemble natural language more closely. Since the preposition is optional it could as well be omitted without changing the way the command is interpreted. However, that also means that using other prepositions that do not make any sense, e.g., 'onto' will not affect the interpretation: the robot will always go *to* the kitchen. But other commands that particularly rely on the interpretation of the preposition could evaluate this part of the sentence. Articles like "the" and "a" and conjunctions like "and" are also recognized during the analysis of the sentence but are ignored, as well as superfluous words like "please". However, it is important to note that these words *may* be typed or spoken, thus making the language interface more human-friendly.

Macro language. A macro language allows the user to define complex command prototypes consisting of simpler commands, e.g.:

```
GO -[preposition] [location] VIA [location]
{
  GO [parameter3]
  GO [optional_parameter1] [parameter2]
}
```

This macro language also allows to create synonyms or translations, i.e., the robot could also be commanded in other languages, e.g., German or French, since their basic grammatical structures are similar to English.

Learning. There are two ways to extend the lexical and syntactical knowledge of the robot. One, the dictionary and the command file may be directly edited since they are plain text files. Two, based on a dialogue conducted by the robot, new words, argument classes and prototypes may be added, and new macro commands can be learned during run time.

3.3 Speech Recognition

Speech recognition is often difficult because of the high level of ambient noise in the environment, e.g., inside a moving car or in office environments where the ambient noise includes various machinery sounds, telephones, moving persons or background conversations. Many methods have been proposed to increase the robustness of speech recognition systems, e.g., training in noisy environments or using multiple microphones, e.g., [Matsui et al. 1997].

Since commercial systems able to cope with these problems are not yet available we require, for the time being, the human to use an ordinary wireless microphone to send his commands to the robot. The used speech recognition engine is a commercial product enabling speaker-independent recognition of continuous speech, which means that users may speak to the system naturally, without pauses between words. This is a very important feature because it allows anybody to communicate with the robot without needing any training with the system. The speech recognition engine generates text strings equivalent to the ones that may be entered via the keyboard.

To render the speech recognition more robust, larger word classes such as [object] have been split into several classes, e.g., [object_to_be_manipulated] and [object_used_for_navigation] which are now used as specific arguments of the command words TAKE or GRASP and MOVE or GO. The fewer the number of words per class and the stricter the syntax, the better the results of the speech recognition will be because fewer hypotheses have to be verified. Also, meaningful results are produced even under noisy conditions.

Contexts. Another important way to increase the robustness of the speech recognition system has been the usage of so-called contexts that contain only those grammatical rules and word lists that are needed for a particular situation. Most parts of robot-human dialogues are situated and built around robot-environment or robot-human interactions, a fact which may be exploited to enhance the reliability and speed of the recognition process. When the robot knows what kind of answers it may expect from the user at a given moment it can switch to a context and disable or enable word lists that are appropriate for the current situation. For example, when the robot asks for confirmation, whether it should execute a certain task or not, the answers will be most likely "yes" or "no" and it would make no sense to expect, and to test, other words. By limiting the set of recognizable words or phrases that can actually be expected, the risk of recognition mistakes is reduced considerably.

However, at any stage in the dialogue a few words and sentences not related to the current context must be available to the user, too. These words are needed to "reset" or bootstrap a dialogue, to "hit the robot's emergency button" and to be able to execute a few other important commands at any time. For example, "Hello, *HERMES*" is used to begin a new dialogue, "Stop" and "Halt" are used for disrupting the robot from its current task, and "Stop listening" and "Continue listening" are used for disabling and enabling the speech recognition engine.

Depending on the prevailing situation and the type of dialogue conducted, various contexts are activated that can be as easy as "Yes" or "No" when *HERMES* expects such an answer, or as complex as a navigation context in which multiple phrases and many words exist to allow complex robot control, especially during supervised learning of environmental features. To teach the robot object and place names, a spelling context has been defined that mainly consists of the international spelling alphabet. Since the spelling alphabet has been optimized for ease of use by humans in noisy environments, such as aircrafts, it should be well suited for robotic applications, too.

3.4 Example: Building a Common Reference Frame

An example of a typical dialogue used to define a place name is given in figure 5. It should be noted how often contexts are switched, depending on the robot's expectations, thus improving the speech recognition considerably.

This dialogue serves also as an example how robots and people can build a common reference frame in their shared working environment. While the robot is guided through the environment or wanders around to explore it, it meets people who would like to share their knowledge about the environment. In fact, sharing knowledge and establishing a common reference frame between a robot and humans is one of the first steps to bring a robot into useful service.

HERMES is in the navigation context which allows, among others, the following instruction:
Human: "Memorize location."
HERMES: "I am taking a panoramic image of the location."
HERMES moves its head 360° to take a panoramic image. Within the image several salient feature are memorized and added as attributes to a topological map of the environment.
HERMES: "I have memorized the location. Do you want to give a name to this location?"
HERMES switches to the YES/NO context.
Human: "Yes"
HERMES: "Please spell!"
HERMES switches to the spelling context.
Human: "Kilo India Tango Charlie Hotel Echo November."
HERMES: "The location is called k i t c h e n - kitchen!"
HERMES switches back to the navigation context.
HERMES: "What else can I do for you?"

Figure 5: Excerpt from a dialogue between a human and *HERMES* to name to a location. Sentences written in italic describe the robot's status or actions.

4 Human-Robot Interaction

4.1 Tactile Sensing

Generally, specific tactile sensors are used for providing robots with a haptic sense. Although we are working towards the development of highly integrated high-resolution tactile sensors to be placed on the gripper surfaces and around the wheelbase, we have found other means of detecting touch events that are helpful in guiding human-robot or robot-environment interaction. We have been able to give the robot a kind of kinesthetic sense by processing angle encoder values and motor currents. Kinesthesia is a "sense mediated by end organs located in muscles, tendons, and joints and stimulated by bodily movements and tensions" [Babcock 1976]. Transferring kinesthetic sensing to the robot for detecting touch events means to detect disturbances on the robot structure that do not result from internal motion requests, but most probably from external circumstances.

Two kinesthetic sensing skills have been developed: one, for detecting touch events or vibrations that occur on any part of the robot structure; two, for detecting unusual external forc-

es during pre-defined robot motions that are, however, unknown to the sensing skill. While the first skill is being used to interact with people in order to shake hands and to hand over or take objects, the second skill allows the robot to smoothly place grasped objects onto other objects. In both cases angle encoder values are sampled at a rate of 1 kHz and low-pass filtered to yield a prediction for the next cycle. If a new angle value deviates significantly from the predicted one, a touch event is signaled to the software module that has requested to detect this touch event (Figure 6).

4.2 Taking, Giving and Placing Objects

Interaction with people and objects requires tactile sensor skills. In combination with motor skills, such as gross arm positioning, objects can be received from, or given to people, or placed onto other objects. Since the robot is not yet skilled enough to visually perceive the current pose of a human hand in order to conform to it, it brings its arm into a configuration where the human user could easily hand over objects or receive them. A slight vibration (touch event) will signal that a human has closed the kinematic chain and is willing to receive or give an object. The robot then closes or opens its gripper, respectively.

To place objects onto other objects, the arm with the grasped object has to be grossly positioned first. Since the perceptual abilities are still limited and do not allow to visually guide the manipulator tip with the grasped object to the required location, the arm is fully stretched out first, and then commanded to move the first elbow joint down while maintaining a horizontal pose of the lower arm. Supervising all arm modules for a touch event will indicate when either the robot arm or the grasped object have touched something. Figure 6 shows an example of the positioning error of the wrist joint during the arm's downward movement. As soon as a touch event is detected, the arm is halted and the gripper is opened, thus relieving immediately the minimal structural stress upon the robot and the object that occurs when the kinematic chain closes.

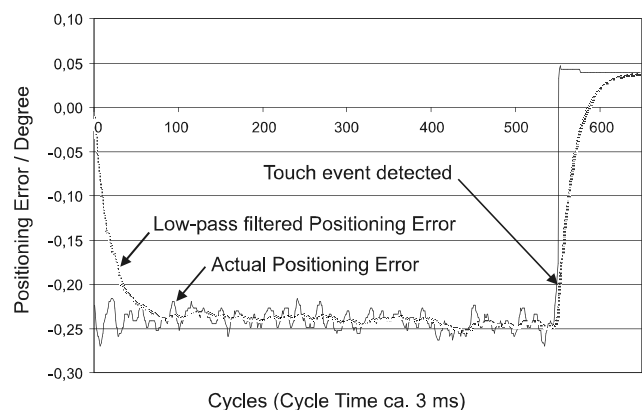


Figure 6: Positioning error (desired position - current position) of the wrist pitch joint during a grasped object's downward movement towards a table. As soon as the object touches the table, the positioning error increases, leading to the detection of a touch event.

5 Experiments and Results

We conducted a number of real-world experiments with the humanoid robot *HERMES* to evaluate the concept presented in the preceding sections. An example that may serve to show the potential of the concept, but also the limitations of the current implementation, is depicted in Figure 7. The corresponding dialogue with a human user is reprinted in the sequel:

Human: "Hello, *HERMES*!"

Being in a state to wait for the next mission to accomplish or a command to be executed, *HERMES*' situation changes as it is addressed with the words "Hello, *HERMES*!". This standard greeting phrase is one of the key phrases *HERMES* is listening to all the time to be able to initialize and to start a dialogue with a human user.

***HERMES*: "Hello! What can I do for you?"**

Human: "Take over glass!"

Simple action- and object-oriented instructions consisting of one command verb and an assigned object are mostly well recognized by the speech recognition system, i.e., they yield a high confidence level. Therefore, the robot will not ask the user for confirmation before actually executing the command. A confirmation is only required if either the confidence level is below a predefined threshold or the instruction would not make any sense in the current context or invoke a number of more complex behaviors that would keep the robot busy for a certain (estimated) amount of time. This mechanism helps to keep the dialog between the robot and the user as fluent as possible.

In this case, the robot checks its data base to figure out what a glass looks like and how it has to be grasped to keep it in an upright and safe pose. Since the robot is not skilled enough to perceive the current pose of the glass (as held by the human) it brings its arm into a configuration where the human can easily hand over the glass. While the gripper is opening it says:

***HERMES*: "Hand over glass, please!"**

Now *HERMES* waits for a touch event that it can recognize with its proprioceptive sensors, specifically an involuntary movement of one or more of its joints that most probably results from external forces. As *HERMES*' encoders yield an

angle resolution of better than 0.05° and are sampled at a rate of up to 1 kHz, very small external forces can be detected and, thus, utilized to signal a touch event. After *HERMES* has successfully grasped the glass, it again waits for further instructions.

***HERMES*: "What else can I do for you?"**

Human: "Place it onto the table!"

This sentence invokes a number of different behaviors. First the robot has to find out what object the user has in mind when referring to "it", then it has to check whether it knows the location of the table and finally it has to approach it. If all parameters are known, the robot will break down the instruction into elementary behaviors and ask for confirmation by repeating what it has understood.

***HERMES*: "I will search for the table, then I will approach the table, then I will place the glass onto the table. Do you want me to do these tasks?"**

Human: "Yes, please!"

Since the shape of the table is known, but its actual position is unknown, the robot starts searching for it (based on the segmentation of the image into obstacle-free and obstacle-occupied regions and contour polygon matching [Bischoff 1999]). In this experiment two tables have been placed in the robot's field of view, so that no intensive search would be necessary and the sample dialogue could continue.

***HERMES*: "I have found two tables. Shall I approach the left or right table?"**

If something unexpected happens or the robot's situation is ambiguous, it informs the user about its current status and then asks for further instructions. The human has to clarify the situation according to the choices given by the robot.

Human: "The left table!"

The speech recognition system only tries to identify the key words "left" or "right" (apart from key phrases to start over or end the current conversation). This improves considerably speech recognition and allows other phrases such as "The left one!" or just the word "left". The final docking to the table is performed using a visual servoing method [Bischoff, Graefe 1998]. The behavior of placing an object onto another object is again performed utilizing the kinesthetic sense described above.



Figure 7: *HERMES* performing a simple service task upon user request: a) greetings and receiving first instructions (via a wireless microphone); b) receiving a glass from the user using kinesthetic sensing; not shown: further conversation with the user as to what to do next (see text); c) and d) approaching left table using a visual servoing method; e) placing glass onto the table.

HERMES: "What else can I do for you?"

Human: "Nothing. Thank you!"

To end the conversation with the robot the user specifies that "nothing" is left that the robot could do for him. Now it is the robot's turn to decide what to do next: waiting for the next mission to accomplish or to obey internal "needs", e.g., to go to the maintenance workshop or battery charger.

This dialogue is a representative example of some of the capabilities of the robot. Natural language recognition is currently restricted to spoken language with a fixed grammar (mostly imperative sentences that have a relatively simple structure). Nevertheless, the robot shows already fairly cooperative and communicative behaviors as appropriate in its actual situation.

6 Conclusions and Outlook

By integrating various sensor modalities including vision, touch and hearing a robot may be built that displays intelligence and cooperativeness in its behavior and communicates in a user-friendly way. This was demonstrated in experiments with a complex robot designed according to an anthropomorphic model. A special kind of behavior-based system architecture has been proposed to control the robot. Its main idea is to select and coordinate the behaviors based on an assessment of the situation being perceived by both the human operator and the robot at a particular moment. This concept places high demands on the robot's sensing and information processing, as it requires the robot to perceive situations and to assess them in real time. A network of microcontrollers and digital signal processors embedded in a single PC, in combination with the concept of skills for organizing and distributing the execution of behaviors efficiently among the processors, is able to meet these demands. Due to the innate characteristics of the situation-oriented behavior-based approach the robot is able to cooperate with a human and to accept orders that would be given to a human in a similar way. Human-robot communication is based on speech that is recognized speaker-independently without any prior training of the speaker. A high degree of robustness is obtained due to the concept of situation-dependent invocations of grammar rules and word lists called "contexts". Human-robot interaction is facilitated by kinesthetic sensing that consists of intelligently processing angle encoder values and motor currents and enables the robot to hand over and take over objects from a human as well as to smoothly place objects onto tables or other objects.

It is a very challenging task to bring together expertise in many diverse disciplines such as electrical and mechanical engineering, computer engineering, and psychology in order to create a robot that closely resembles a human not only in size and shape but also in sensory and motor skills. Although we are very far from creating human-like skills and intelligence in an embodied form, methods developed for humanoids could as well enhance current service robots and

lead to the development of personal robots in the future. In contrast to today's specialized service robots these personal robots could well be used in many different environments (domestic, public and industrial) for a variety of tasks (e.g., elderly care, helping handicapped people, assistance in factories).

Acknowledgments

We thank Mr. Nicolas Dalstein (ENSI, Caen, France) for writing major parts of the command interpreter, and especially Prof. Dr. Volker Graefe for the many fruitful discussions we had.

References

- Arkin, R. C. (1998).** Behavior-Based Robotics. MIT Press, Cambridge, MA, 1998.
- Babcock, P. (1976):** Webster's Third New International Dictionary of the English Language, G. & C. Merriam Company, Springfield, MA, 1976.
- Bergener, T.; Bruckhoff, C.; Dahm, P.; Janßen, H.; Joublin, F.; Menzner, R. (1997).** Arnold: An Anthropomorphic Autonomous Robot for Human Environments. In: H.-M. Groß (Hrsg.): Fortschrittsberichte VDI, Reihe 8, Nr. 663, Workshop SOAVE'97, Ilmenau, September 1997, pp 25-34.
- Bischoff, R. (1998).** Design Concept and Realization of the Humanoid Service Robot *HERMES*. In A. Zelinsky (ed.): Field and Service Robotics. Springer, London 1998, pp. 485-492.
- Bischoff, R. (1999).** Advances in the development of the humanoid service robot *HERMES*. Second International Conference on Field and Service Robotics. Pittsburgh, PA, Aug. 1999, 156-161.
- Bischoff, R.; Graefe, V. (1998).** Machine Vision for Intelligent Robots. IAPR Workshop on Machine Vision Applications. Maku-hari/Tokyo, November 1998, pp. 167-176.
- Bischoff, R.; Graefe, V. (1999).** Integrating Vision, Touch and Natural Language in the Control of a Situation-Oriented Behavior-Based Humanoid Robot. IEEE Conference on Systems, Man, and Cybernetics, October 1999, to appear.
- Brooks, R. A.; Stein, L. A. (1993).** Building Brains for Bodies. A.I. Memo No. 1439, Massachusetts Institute of Technology, Boston, August 1993.
- Graefe, V.; Bischoff, R. (1997).** A Human Interface for an Intelligent Mobile Robot. 6th IEEE Intern. Workshop on Robot and Human Communication, Sendai, Japan, Sept. 1997, pp. 194-197.
- Honda (1997).** Honda Introduces "Human" Robot. http://www.honda.co.jp/home/hpr/e_news/robot/index.html
- Konno, A.; Nagashima, K.; Furukawa, R.; Nishiwaki, K.; Noda, T.; Inaba, M.; Inoue, H. (1997).** Development of the Humanoid Robot Saika. Proc. of IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems, IROS '97, Sept. 1997, pp. 805-810.
- Matsui, T.; Asoh, H.; Asano, F. (1997).** Map Learning of an Office Conversant Mobile Robot, Jijo-2, by Dialogue-Guided Navigation. Proc. of the Intern. Conference on Field and Service Robotics. Canberra, Australia, December 1997, pp. 230-235.
- Schraft, R. D.; Schmierer, G. (1998).** Serviceroboter – Produkte Szenarien, Visionen. Springer-Verlag, Berlin, 1998 (in German).
- Yamaguchi, J.; Takanishi, A. (1997).** Design of Biped Walking Robot Having Antagonistic Driven Joints Using Nonlinear Spring Mechanism. Proc. of IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems, IROS '97, Sept. 1997, pp. 251-259.