In Proceedings of 1st International Conference on Dextrous Autonomous Robots and Humanoids (darh2005)
Yverdon-les-Bains - Switzerland, paper nr. 3.2, May 2005.

Using Handheld Computers to Control Humanoid Robots

Sven Behnke, Jürgen Müller, and Michael Schreiber

University of Freiburg, Department of Computer Science
Georges-Köhler-Allee 52, 79110 Freiburg, Germany
{behnke, jmuller, schreibe}@informatik.uni-freiburg.de

Abstract

Small humanoid robots available today often lack computing power and vision sensors. They frequently consist only of servos and microcontroller boards.

We propose to use off-the-shelf handheld computers, Pocket PCs and ultra-portable PCs, to make them autonomous. These computers are lightweight, compact, robust, affordable, and have many interfaces. It is not hard to attach them to an existing robot, to interface them to the microcontrollers, and to add a camera to them.

The handheld computers have ample computing power to run image processing, self-localization, behavior control, and communication onboard a robot. We used the proposed approach to augment the robots RoboSapien, KHR-1, and Toni. The paper reports experiences made with these autonomous humanoid robots in the RoboCup soccer domain.

1 Introduction

Humanoid robots are enjoying increasing popularity as a research tool, since their anthropomorphic body allows the investigation of human-like motion and multimodal communication.

Some impressive humanoid robots, like Asimo [1] and Qrio [2], have been developed by large companies. Unfortunately, these robots are not available to researchers outside industry labs.

Some simpler designs are available commercially, e.g. from Vstone [3] or from Kondo [4]. These small robots are constructed from up to 22 servos. They contain microcontroller boards to generate target signals for the servos, and have a RS232 serial interface. Another commercial humanoid robot is RoboSapien [5], which has been developed for the toy market.

All of these robots lack sufficient user-programmable computing power and a vision sensor to be used as autonomous humanoid robots. The same applies to many humanoid robots constructed in research labs.

In order to make their robots autonomous, most researchers turn to industrial computers, such as PC104 or other single-board computers. These computers lack a display and possibilities for user input, need a power supply, wireless communication, housing, and cooling. While it is possible to add all these components in order to build a complete system, the modular approach leads to a suboptimal design. Reliability and weight/performance ratio are compromised. In single units, industrial computers also tend to be pricy.

Another approach to make humanoid robots autonomous is to use a handheld computer, such as a Pocket PC or an ultra-portable PC. These computers are lightweight, compact, robust, affordable, and have many interfaces. It is not hard to attach them to an existing robot, to interface them to microcontrollers, and to add a camera to them.

In this paper, we describe the realization of the latter approach with two different handheld computers and three different humanoid robots.

The remainder of the paper is organized as follows. In the next section, we review some of the related work. Section 3 describes the handheld computers used. In Section 4, we introduce three humanoid robots and detail the interface to the handheld computers. Computer vision and self-localization are described in Section 5. The behavior control software for these robots is covered in Section 6. Wireless communication and other infrastructure components are presented in Section 7. The paper concludes with a discussion of the experiences made with the augmented humanoid robots in the RoboCup soccer domain.

2 Related Work

The above mentioned problems of industrial computers are not unique to the use in humanoid robots. Some researchers used off-the-shelf PDAs to control their wheeled robots [6, 7]. One of the best known PDA projects is the Palm Pilot Robot Kit [8, 9], developed at CMU.

A PDA has also been used to control the Robota and DB humanoid robots [10], which are not mobile. The humanoid robot GuRoo, constructed at the University of Queensland [11], is controlled by a Compaq iPAQ Pocket PC. Its cameras are attached to a dedicated vision board, which performs color segmentation and basic computer vision algorithms.

Several wheeled robots, e.g. [12, 13], are controlled by subnotebook PCs, which are interfaced to digital cameras via FireWire or USB. A subnotebook has also been used to control the humanoid robot Alpha, constructed in our lab [14].

3 Handheld Computers

Depending on the size of the humanoid robot and its allowed payload, we used either a Pocket PC or a ultra-portable PC to make it autonomous.

3.1 Pocket PCs

For the smaller robots it is essential to find a lightweight computer, such as a Pocket PC and a tiny camera. Among the many available Pocket PCs we selected the FSC Pocket Loox 720, which weighs only 170g, including the battery. Its size is 12.2×7.2×1.5cm. It features a 520MHz XScale processor PXA-272, 128MB RAM, and 64MB flash memory. The Pocket Loox 720 has a 3.6" touch-sensitive display with VGA resolution (640×480). It has a built-in microphone, speakers, and an integrated 1.3MPixels color camera. For connectivity the Pocket Loox 720 is equipped with Bluetooth, wireless LAN, a RS232 serial interface, and an IrDA infrared interface. Its replaceable Lithium-Ion battery has a capacity of 6Wh, sufficient for several hours of operation. Finally, there are a SD-slot and a CF-slot available.

Because the API for the internal camera has not been released, we added a Lifeview FlyCAM CF camera with 1.3MPixels to the Pocket Loox 720. In order to allow for a large field-of view, we replaced the original lens with an ultra-wide-angle lens. The field of view is now about 150×112°. The CF camera can capture RGB-images in different resolutions. At 320×240 the maximum frame rate is 5fps.

3.2 Ultra-Portable PCs

While the weight/performance ratio of Pocket PCs is outstanding, their absolute performance is lower than the performance of PCs. PCs have decreased in size over the last years and have recently almost reached the form factor of Pocket PCs. Two ultra-portable handheld PCs are currently available on the market: OQO 01 and Sony Vaio U-Series. We selected the Vaio, because it is equipped with a faster processor than the OQO.

As shown in Fig. 1(a), the U750P is small enough to fit into the trunk of our humanoid robot Toni. Its size is 16.8×10.9×2.6cm and it weighs 550g, including batteries. The U750P features an ultra-low voltage 1.1GHz Pentium M 733 processor, 512MB RAM, and 20GB harddrive. Its 5" touch-sensitive display has SVGA resolution (800×600). Speakers and a headphone jack are integrated. The U750P has a CF-slot and a memory stick slot. One USB2.0 port and wireless LAN provide connectivity. The 20Wh replaceable Lithium-Ion battery is sufficient for 1.5-3.0 hours of operation. The U750P comes with a foldable keyboard and a docking station with more interfaces, including Ethernet, FireWire, and more USB ports.

In order to use the U750P without the docking station, we added a 4-port mini-USB-hub to it. One port connects to a USB-RS232 converter. Two other ports interface ultra-wide-angle USB cameras, also visible in Fig. 1(a). They consist of webcam electronics, a 1/3"CCD imager, and a door viewer lens. Toni can see almost all objects around it simultaneously when one of these cameras is pointed forwards and the other backwards. Two images captured at the same time are shown in Fig. 1(b). The cameras deliver up to 30fps at resolutions up to 640×480.

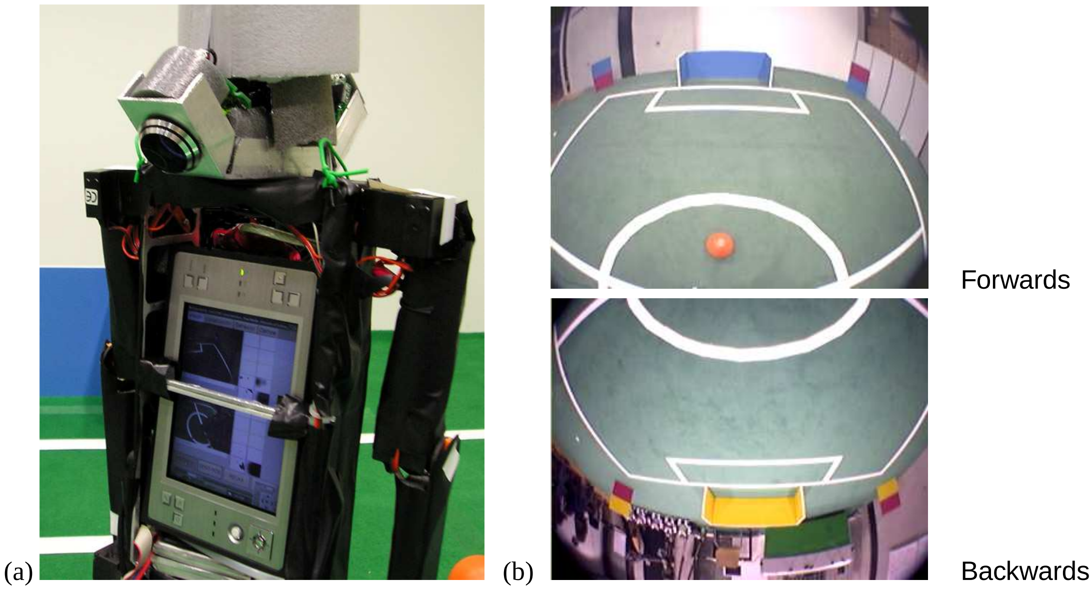


Figure 1: Toni with a small PC and two ultra-wide-angle cameras: (a) Picture of the robot's back and neck; (b) Pictures captured simultaneously from the two cameras.

4 Robots

We augmented three different humanoid robots with computing power and cameras: Kondo, RoboSapien, and Toni.

4.1 Kondo KHR-1

The first robot, shown in Figure 2(a), is the KHR-1 construction kit made by Kondo [4]. In its original configuration the Kondo robot is driven by 17 servos: 5 in each leg, 3 in each arm, and one used as head. It is 34cm tall and has a total weight of 1.2kg. We do not use the head servo and instead augmented the robot with a Pocket PC equipped with a wide-angle camera. Recently, we added two more DOFs in its feet.

The Pocket PC interfaces the servo control boards via a RS232 serial line. It sends target positions for the individual joints at a rate of 50Hz. The Kondo robot is powered by NiCd batteries.

4.2 RoboSapien

Figure 2(b) shows an augmented RoboSapien. The RoboSapien base has been developed by M. Tilden and is marketed by WowWee for the toy market [5]. It is driven by 7 DC motors and powered by batteries located in its feet. The low center of mass makes RoboSapien very stable. Using three motors it can walk dynamically with two speeds in sagittal direction. It can also turn on the spot. The other four motors move its arms.

The original RoboSapien is controlled by the user with a remote control. We added a Pocket PC and a wide-angle CF-camera. The Pocket PC sends motion commands to the robot base at a rate of up to 4Hz. We use a learning remote software (UltraMote) to convert Windows messages, emitted by our application, into infrared commands.

4.3 Toni

Figure 2(c) shows the robot Toni, which has been constructed in our lab. Toni is 74cm tall and has a total weight of 2.2kg, including a Pocket Loox 720 and a wide-angle CF camera, or 2.7kg, including an U750P PC and two USB-cameras.

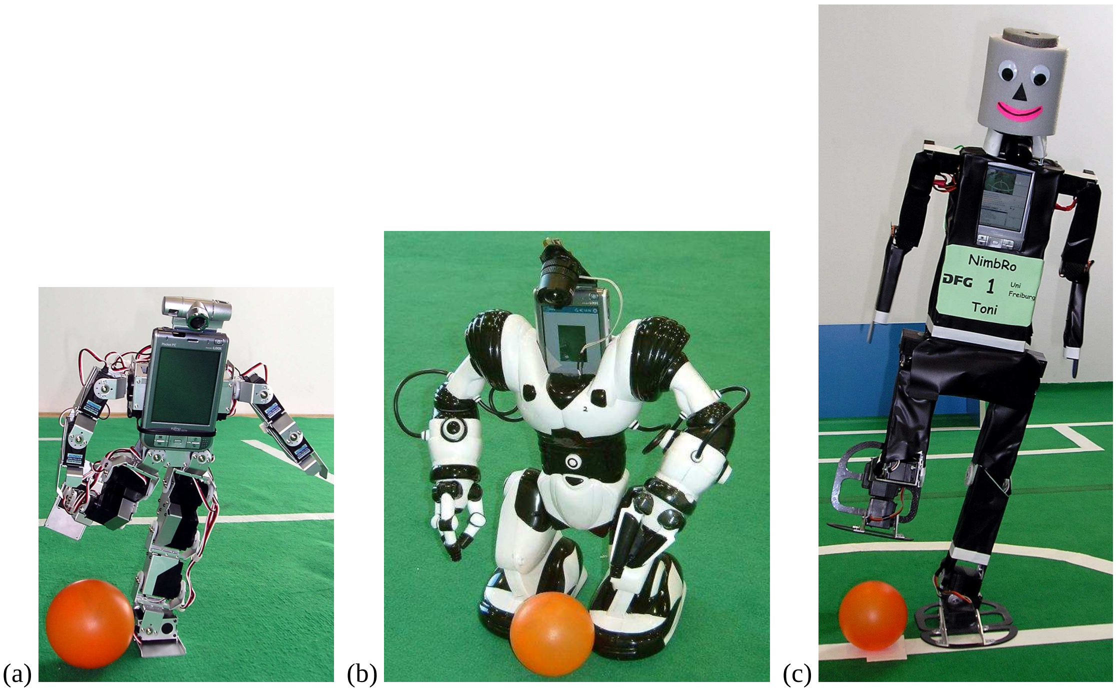


Figure 2: (a) augmented Kondo KHR-1; (b) augmented RoboSapien; (c) Toni with Pocket PC.

The robot is driven by 18 servos: 6 in each leg and 3 in each arm. Its mechanical design focused on human-like proportions and a light weight. In order to be able to over-extend the leg, we added a toes joint to its foot plate. This allows it to dynamically walk in a human-like manner with a straight stance leg and toes bending before toes-off.

Toni is fully autonomous. It is powered by Lithium-polymer batteries. Three HCS12 microcontrollers generate target signals for the servos and read back the servo positions and duties. They also interface an attitude sensor consisting of two accelerometers and two gyros. The microcontrollers communicate with each other via a CAN bus and with the main computer (Pocket PC/ultra-portable PC), located in its trunk, via a RS232 serial line. The main computer sends target positions for the individual joints at a rate of 167Hz. It reads back preprocessed sensor data.

5 Perception

Our robots need information about themselves and about the world around them to act successfully.

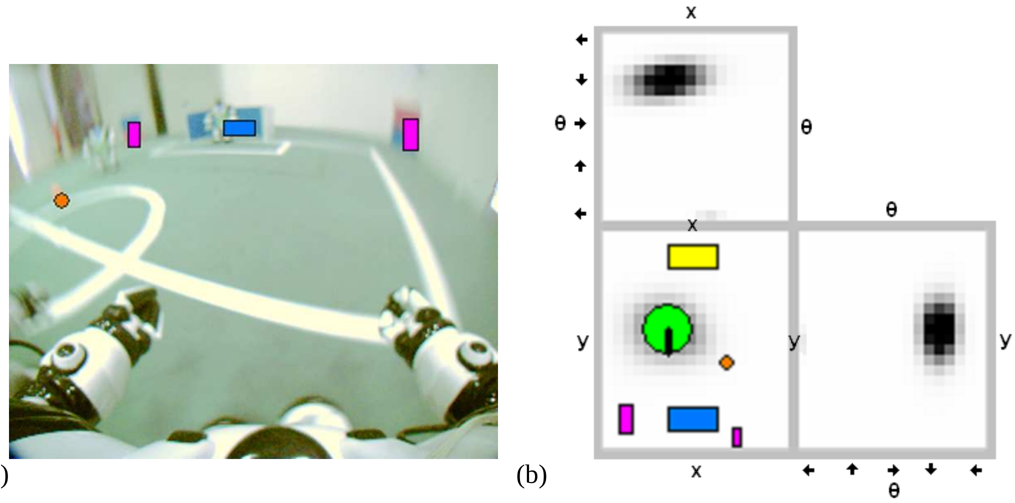
We fuse the accelerometer and gyro readings of Toni to obtain an estimate of its tilt and keep track of its leg joint angles and motor duties by interpreting its potentiometer voltages.

The only source of information about the environment of our robots is their camera. On the RoboCup soccer field, the wide-angle CF-cameras allow seeing the ball at the robots feet, the goal, and some of the color markers around the field simultaneously (see Fig. 3(a)).

Our computer vision software detects these key objects based on their color and estimates their coordinates in an egocentric frame (distance to the robot and angle to its orientation). This suffices for many relative behaviors, like positioning behind the ball facing the goal.

To implement global team behaviors, such as kick-off, we need the robot coordinates in an allocentric frame (position on the field, orientation). We estimate these using a probabilistic Markov localization method that integrates egocentric observations and motion commands over time. As proposed by Fox, Burgard, and Thrun [15] this method uses a three-dimensional grid (x, y, θ) to represent the likelihood of robot poses. Three 2D-projections of this grid are shown in Figure 3(b).

Based on robot localization, we compute relative coordinates for objects currently outside the robot's field of view. Localization is also used to fuse multiple local robot views into a global team view, e.g. by



(a) (b)

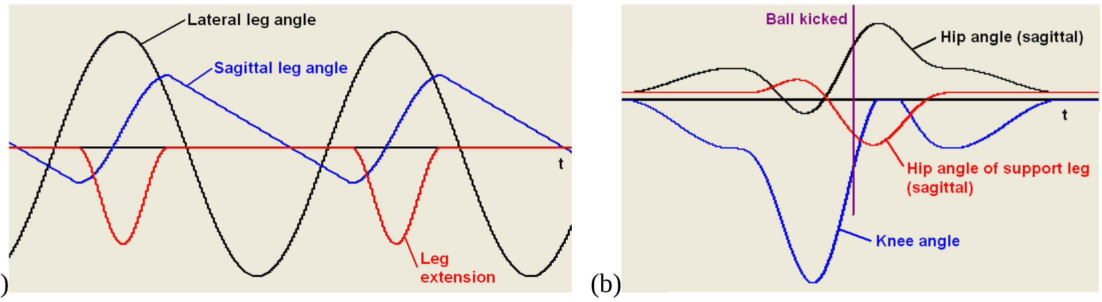
Figure 3: (a) Image captured from RoboSapien's perspective while it was walking on a RoboCup soccer field. Detected objects: goal (blue horizontal rectangle), ball (orange circle), and markers (magenta vertical rectangles); (b) Three two-dimenensional projections of the grid representing the likelihood of robot poses (x, y, θ). The green circle is drawn at the estimated robot location (x, y). The black line represents its estimated orientation θ. The detected objects are drawn relative to the robot.

integrating ball observations of multiple robots over time.

6 Behavior Control

We control our robots using a framework that supports a hierarchy of reactive behaviors [16]. For the servo-based robots, we generate target-positions for the individual joints at a high rate.

To abstract from these many degrees of freedom, the next higher level generates targets for body parts, such as leg extension and leg angle (between hip and line from hip joint to ankle joint). On this layer we implemented dynamic walking. Fig. 4(a) shows some trajectories generated for Toni while walking with small steps in forward direction.



(a) (b)

Figure 4: Trajectories generated for Toni: (a) Dynamic Walking. Leg angles and leg extension of a leg while walking forward with small steps. The other leg moves in the same way, with a phase offset of π. (b) Kicking. Joint angles of the knee of the kicking leg and the sagittal hip joints of both legs.

Toni's maximum walking speed is about 20cm/s. The robot is not only able to walk forward and backward, but it can also walk to the side and turn on the spot. By blending these gait directions, we generate omnidirectional walking. We used this interface to implement higher-level behaviors, like approaching the ball and dribbling. In addition to walking we implemented a kicking behavior for Toni (see Fig. 4(b)). Toni smoothly accelerates and decelerates the kicking foot. When the ball is hit, the angular velocities of its kicking knee and both hips add up in forward direction. After kicking, the ball rolls for about 2m on carpet and Toni returns to a stable stand.

Kondo is able to walk at a speed of up to 10cm/s. We adapted the omnidirectional walking and the kicking behaviors of Toni for it. Currently, we are working on automatically optimizing Kondo's gait pattern. We also plan to investigate getting-up behaviors with Kondo.

Since we cannot change the gait of RoboSapien, behavior control for it can focus on higher-level issues. We implemented a variety of soccer behaviors for a team of augmented RoboSapiens and played multiple test games. In a scoring test (empty field, 10 different ball positions), we compared the performance of the autonomous RoboSapien to the same robot controlled via the original remote unit by a human operator. While both achieved a 10/10 success rate, the autonomous RoboSapien needed on average about 175% of the time of the human operated robot (170s vs. 97s).

7 Infrastructure

All three augmented humanoid robots are equipped with wireless network adapters. We use the UDP protocol to transmit debug information to an external computer, where it is logged and visualized. This computer is also used to fuse local views into a team view. For soccer games with multiple RoboSapiens we are using it to compute team behaviors, such as the assignment of roles to the individual soccer players. We are also using the wireless network for transmitting the game state (kickoff, penalty ...) to the robots and to remotely control the robots during tests, e.g. with a joystick.

In order to be able to design behaviors without access to the real hardware, we implemented a physics-based simulation for all three robots. This simulation is based on the Open Dynamics Engine [17].

8 Conclusions

The use of highly integrated off-the-shelf handheld computers made it easy to augment three different humanoid robots to autonomous robots. The selected Pocket PC and the ultra-portable PC are lightweight, compact, robust, affordable, and have many interfaces.

We described how we attached the computers to the robots and how we interfaced them via an RS232 serial line or infrared. The API for capturing images and sending commands to RoboSapien is available on our website. We also provided the code for controlling the Kondo robot to other groups.

We added an ultra-wide-angle CF-camera to the Pocket PCs and two ultra-wide-angle USB-cameras to the PC. The Pocket PCs are fast enough to analyze the images (320×240, RGB) and to compute probabilistic self-localization at a rate of 4Hz. Behavior control and wireless communication run in parallel at a rate depending upon the robot (RoboSapien 4Hz, Kondo 50Hz, Toni 167Hz).

The PC is more powerful and can analyze the images of two cameras at a higher rate. While the total number of frames per second (320×240, RGB) exceeds 33, the U750P is utilized only 75%. This leaves resources for more demanding behavior control algorithms, such as path planning.

Due to the low price of the augmented RoboSapien, we were able to augment five of these robots. In a lab project we prepared for 4 vs. 4 demonstration games against the Brainstormers Osnabrück, which took place at the RoboCup German Open in Paderborn (April 2005). These first humanoid robot soccer games were exciting and drew many spectators.

The servo-driven robots Kondo and Toni demonstrated penalty kicks against the Darmstadt Dribblers at this occasion as well. Our robots were able to approach the ball and to kick it towards the goal. If the ball did not pass the goal line, the robots were able to approach and kick it a second time.

Several videos of the augmented robots can be found at our project web pages: http://www.NimbRo.net.

Acknowledgments

Funding for the project is provided by Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under grant BE 2556/2-1.

Tobias Langner worked on the camera interface for the Pocket PC. Jörg Stückler implemented the ODE simulations. Thorsten Kramer, Julio Pastrana, Johannes Schwenk, and Konstantin Welke implemented behaviors for the robots.

The RoboCup team FU-Fighters provided the sources of a framework for implementing a hierarchy of reactive behaviors.

References

[1] Honda. ASIMO. http://world.honda.com/asimo.

[2] Sony. Dream Robot QRIO. http://www.sony.net/qrio.

[3] Vstone Co., Ltd. http://www.vstone.co.jp.

[4] Kondo Kagaku Co., Ltd. Khr-1. http://www.kondo-robot.com.

[5] Mark W. Tilden. Neuromorphic robot humanoid to step into the market. *The Neuromorphic Engineer*, 1(1):12, 2004.

[6] Doug Williams. *PDA Robotics*. McGraw-Hill, 2003.

[7] Kevin Mukhar, Dave Johnson, Kevin Mukhar, and Dave Johnson. *The Ultimate Palm Robot*. McGraw-Hill, 2003.

[8] CMU. Palm Pilot Robot Kit. http://www-2.cs.cmu.edu/∼reshko/pilot.

[9] Greg Reshko, Matthew Mason, and Illah Nourbakhsh. Rapid prototyping of small robots. Technical Report CMU-RI-TR-02-11, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2002.

[10] Sylvain Calinon and Aude Billard. PDA interface for humanoid robots. In *Third IEEE International Conference on Humanoid Robots (Humanoids 2003)*, 2003.

[11] Gordon F. Wyeth, Damien Kee, Mark Wagstaff, Nathaniel Brewer, Jared Stirzaker, Timothy Cartwright, and Bartek Bebel. Design of an autonomous humanoid robot. In *Proceedings of the Australian Conference on Robotics and Automation (ACRA 2001), Melbourne*, 2001.

[12] Sven Behnke, Alexander Gloye, Felix von Hundelshausen, and Raul Rojas. FU-Fighters 2002 (Middle Size). In *Team descriptions, Middle Size League, RoboCup 2002 Competition, Fukuoka, Japan*, 2002.

[13] Fraunhofer AIS. VolksBot. http://www.ais.fraunhofer.de/be/volksbot.

[14] Sven Behnke, Norbert Mayer, Jürgen Müller , and Michael Schreiber. Nimbro 2004 team description. In *Team descriptions, Humanoid League, RoboCup 2004 Competition, Lisbon*, 2004.

[15] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[16] Sven Behnke and Raul Rojas. A hierarchy of reactive behaviors handles complexity. In M. Hannebauer, J. Wendler, and E. Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, pages 125–136. Springer, 2001.

[17] Russel Smith. Open Dynamics Engine. http://opende.sourceforge.net.