NimbRo KidSize 2007 Team Description

Sven Behnke, Michael Schreiber, Jörg Stückler, Hauke Strasdat, and Konrad Meier

Albert-Ludwigs-University of Freiburg, Computer Science Institute
Georges-Koehler-Allee 52, 79110 Freiburg, Germany
{ behnke | schreibe | stueckle | strasdat | meierk } @ informatik.uni-freiburg.de
http://www.NimbRo.net

Abstract. This document describes the RoboCup Humanoid League team NimbRo KidSize of Albert-Ludwigs-University Freiburg, Germany, which is qualified for the competition to be held in Atlanta in July 2007. Our team uses self-constructed robots for playing soccer. The paper describes the mechanical and electrical design of the robots. It also covers the software used for perception, behavior control, communication, and simulation.

1 Introduction

The project NimbRo – Learning Humanoid Robots is running at Albert-Ludwigs-University of Freiburg, Germany, since 2004. Our KidSize team participated with success at last year's RoboCup Humanoid League competition in Bremen, Germany. Our robots kicked penalties very reliably. In the Penalty Kick competition they scored in 31 of 34 attempts. They won the Penalty Kick Final 8:7 against Team Osaka (Fig. 10(a)). They also reached the final in the 2 vs. 2 soccer games (Fig. 10(b)). This exciting game was won by the titleholder, Team Osaka in the extra time, after a 4:0 lead for NimbRo and a 4:4 draw at the end of the regular playing time. Our KidSize robot Gerd was one of two robots able to walk across the rough terrain. The NimbRo KidSize robots also scored in the passing challenge. Overall, NimbRo KidSize came in second in the Best Humanoid ranking.

Our KidSize robots won the RoboCup German Open 2007 competition, which took place in April at Hannover Messe. They scored a total of 41:0 goals in four 2 vs. 2 soccer games. Our robots met Darmstadt Dribblers in the final and won the game 15:0. The NimbRo KidSize robots also won the technical challenges in Hannover, which consisted of obstacle avoidance, passing, and dribbling.

For the RoboCup 2007 competition, we prepare not only for the 2 vs. 2 soccer games and the technical challenges, but also for 3 vs. 3 soccer demonstration games. This document describes the current state of the project as well as the intended development for the 2007 RoboCup competitions. It is organized as follows. In the next section, we describe the mechanical design of the robots. Sec. 3 details the robot electronics. The perception of the internal robot state and the situation on the field is covered in Sec. 4. Sec. 5 describes the infrastructure

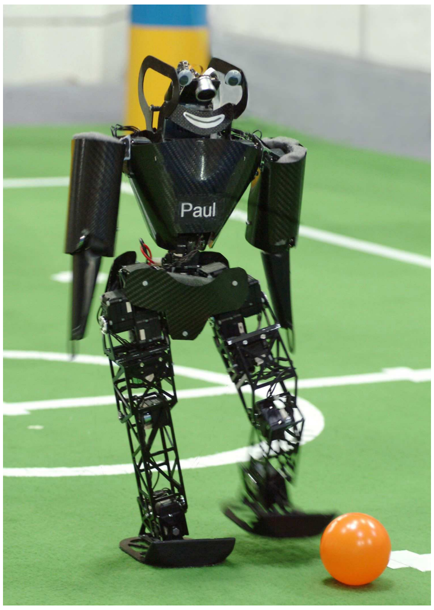


Fig. 1. NimbRo 2006 KidSize robot Paul.

needed to support a team of soccer playing robots. The generation of soccer behaviors in a hierarchy of agents and time-scales is explained in Sec. 6.

2 Mechanical Design

Fig. 1 shows Paul, one of our 2006 KidSize robots. As can be seen, the robot has human-like proportions. Its mechanical design focused on simplicity, robustness, and weight reduction. The KidSize robots have a height of 60cm and weigh only 2.9kg, including batteries. They are driven by 24 Dynamixel actuators: 8 per leg, 3 in each arm, and two in the trunk. For the leg and the trunk joints, we use the DX-117 actuators (66g, 37kg·cm). Three orthogonal axes constitute the 3DOF hip joint. For the hip pitch and roll axes, we use two of these actuators in a parallel master-slave configuration. This doubles the torque and lowers operating temperatures. Two orthogonal servos form the 2DOF ankle joint and one servo drives the knee joint. The trunk joints are in the pitch and yaw axes. The arms do not need to be as strong. They are powered by DX-113 actuators (58g, 10.2kg·cm). Two orthogonal servos constitute the shoulder joint and one servo drives the elbow joint. The skeleton of the robot is constructed from aluminum extrusions with rectangular tube cross section. We removed all material not necessary for stability. The feet, the forearms, and the head are made from sheets of elastic carbon composite material.

3 Electronics

The Dynamixel actuators have a RS-485 differential half-duplex interface. Each robot is equipped with a HCS12 microcontroller board, which manages the detailed communication with all Dynamixels. The Dynamixel actuators have a flexible interface. Not only target positions are sent to the actuators, but also parameters of the control loop, such as the compliance. In the opposite direction, the current positions, speeds, loads, temperatures, and voltages are read back. In addition to these joint sensors, each robot is equipped with an attitude sensor, located in the trunk. It consists of a dual-axis accelerometer (ADXL203, ± 1.5g) and two gyroscopes (ADXRS 300, ± 300 °/s). The sensor signals are digitized with A/D converters of the HCS12. The microcontroller communicates with the Dynamixels via RS-485 at 1MBaud and with a main computer via a RS-232 serial line at 115KBaud. Every 12ms, target positions and compliances for the actuators are sent from the main computer to the HCS12 board, which distributes them to the actuators. The HCS12 sends the preprocessed sensor readings back.

We use a Pocket PC as main computer, which is located in the upper part of the robot [2]. The FSC Pocket Loox 720 has a weight of only 170g, including the battery. It features a 520MHz XScale processor, a touch-sensitive display with VGA resolution, wireless LAN, a RS-232 serial interface, and an integrated 1.3 MPixel camera. This computer runs behavior control, computer vision, and wireless communication. We took the integrated camera out of the Pocket PC and connected it via an extension cable. Located above the Pocket PC, it looks in forward direction through a wide-angle converter. 640×480 images can be captured at 15fps using DMA. In addition, a Lifeview FlyCam CF 1.3M that has been fitted to an ultra-wide-angle lens is looking in backward direction. Our soccer robots are powered by high-current Lithium-polymer rechargeable batteries, which are located in their lower back. Four Kokam 910mAh cells last for about 25 minutes of operation.

4 Perception

Our robots need information about themselves and the situation on the soccer field to act successfully.

4.1 Proprioception

The readings of accelerometers and gyros are fused to estimate the robot's tilt in roll and pitch direction. The gyro bias is automatically calibrated and the low-frequency components of the tilt estimated from the accelerometers are combined with the integrated turning rates to yield an estimate of the robot's attitude that is insensitive to short linear accelerations. As described above, joint angles, speeds, and loads are also available. Temperatures and voltages are monitored to notify the user in case of overheating or low batteries.

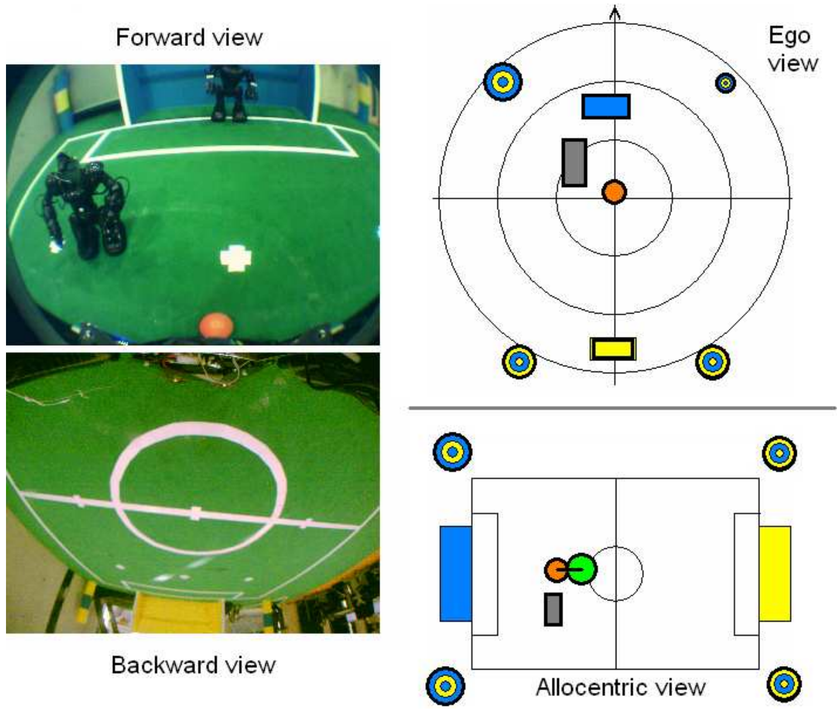


Fig. 2. Left: Images of the two cameras mounted on the robot. Upper right: Egocentric coordinates of key objects (ball, goals, corner poles, obstacle) detected in the image. Lower right: Localization of the robot, the ball, and the obstacle on the soccer field.

4.2 Visual Object Detection

The only source of information about the environment for our robots is their camera. The wide field of view of the cameras allows the robots to see their own feet and objects above the horizon at the same time (see left part of Fig. 2). Our computer vision software detects the ball, the goals, the corner poles, and other players based on their color in YUV space. Using a look-up table, the colors of individual pixels are classified into color-classes that are described by ellipsoids in the UV-plane. In a multistage process we discard insignificant colored pixels and detect colored objects. We estimate their coordinates in an egocentric frame (distance to the robot and angle to its orientation), based on the inverted projective function of the camera. We correct first for the lens distortion and invert next the affine projection from the ground plane to the camera plane. The estimated egocentric coordinates of the key objects are illustrated in the upper right part of Fig. 2. Here, the objects detected by both cameras are fused, based on their confidence. The objects are also merged with previous observations, which are adjusted by a motion model, if the robot is moving. This yields a robust egocentric world representation.

4.3 Self-Localization

The relative coordinates suffice for many relative behaviors like positioning behind the ball while facing the goal. To keep track of non-visible goals or to communicate about moving objects with other team members, we need the robot coordinates in an allocentric frame ((x, y)-position on the field and orientation θ). We solve self-localization by triangulation over pairs of landmark observations, i.e. detected goals and corner poles. When observing more than two landmarks, the triangulation results are fused based on their confidence. Again, the results of self-localization are integrated over time and a motion model is applied. The lower-right image of Fig. 2 illustrates the resulting allocentric representation.

5 Infrastructure

To support a team of soccer playing robots some infrastructure components are needed.

• **Wireless Communication:** The Pocket PCs of our robots are equipped with wireless network adapters. We use the wireless communication to transmit debug information to an external computer via UDP, where it is logged and visualized. This allows the behavior engineers to keep track of the perceptions and the chosen actions. The wireless network is also used for transmitting the game state (kickoff, penalty, etc.) from the external PC to the robots. The robots communicate with each other to share perceptions and to negotiate roles.

• **Simulation:** In order to be able to design behaviors without access to the real hardware, we implemented a physics-based simulation for the robots. This simulation uses the Open Dynamics Engine [5]. It was very helpful for the design of getting up motions, goalie motions, and team behaviors. The simulator is also used to develop learning algorithms, which are then applied to the real robot.

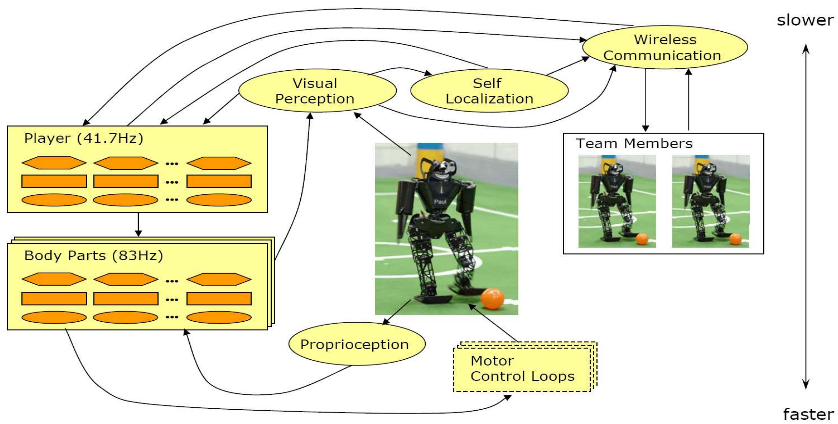


Fig. 3. Overview of the behavior control architecture.

6 Behavior Control

6.1 Hierarchy of Reactive Behaviors

We control the robots using a framework that supports a hierarchy of reactive behaviors [3]. This framework allows for structured behavior engineering. Multiple layers that run on different time scales contain behaviors of different complexity. The framework forces the behavior engineers to define abstract sensors that are aggregated from faster, more basic sensors. One example for such an abstract sensor is the robot's attitude that is computed from the readings of accelerometers and gyros. Abstract actuators allow higher-level behaviors to configure lower layers in order to eventually influence the state of the world. One such abstract actuator is the desired walking direction, which configures the gait engine, described below, implemented in the lower control layers.

The framework also supports an agent hierarchy. For our soccer robots, we use four levels of this hierarchy: individual joint – body part – entire robot – team. This structure restricts interactions between the system variables and thus reduces the complexity of behavior engineering. Fig. 3 illustrates the interactions of the different parts of the behavior control framework.

The lowest level of this hierarchy, the control loop within the Dynamixel actuators, has been implemented by Robotis. It runs at a high frequency. We monitor targets, actual positions, speeds, and loads. At the next layer, we generate target positions for the individual joints of a body part at a rate of 83.3Hz. We make sure that the joint angles vary smoothly. To abstract from the individual joints, we implemented here, for example, an interface that allows to change leg extension, leg angle, and foot angle.

6.2 Omnidirectional Walking

On the next higher level, we use this leg interface to implement omnidirectional walking. Shifting the weight from one leg to the other, shortening of the leg not needed for support, and leg motion in walking direction are the key ingredients of this gait. In contrast to the low-frequency gait of our 2005 robots [1], we were able to increase the step frequency significantly to 3.45Hz. We also use feedback from the rate gyros to stabilize the robots. Fig. 4 illustrates the gait for the tree principal directions. Note that the leg is not only shortening during swing, but also in the middle of the stance phase.

The three basic walking directions can be smoothly combined. The robots are able to walk in every direction and to change their heading direction at the same time. The gait target vector (v_x, v_y, v_θ) can be changed continuously while the robot is walking. This makes it possible to correct for deviations in the actual walking direction and to account for changes in the environment by using visual feedback. When using this flexible gait, the maximal forward walking speed of the robots is approx. 25cm/s, but they walk slower in the vicinity of obstacles and the ball.

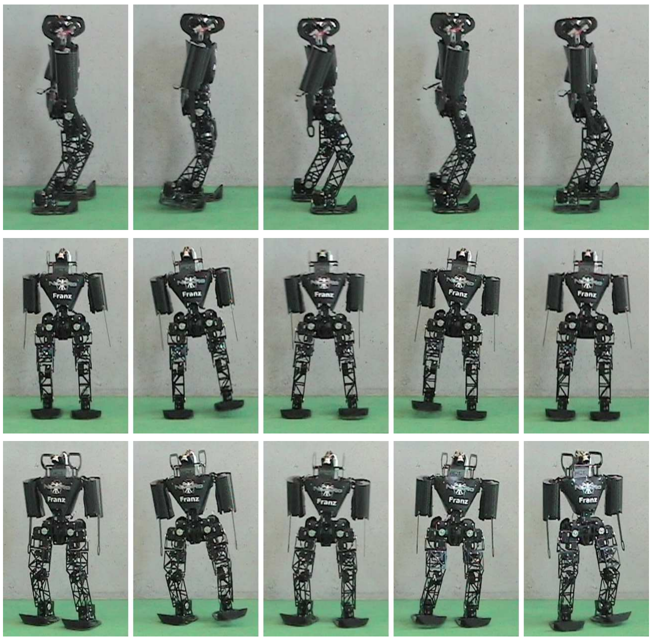


Fig. 4. Omnidirectional walking: forward, lateral, and turning on the spot.

6.3 Fall Avoidance, Controlled Fall, Getting-up

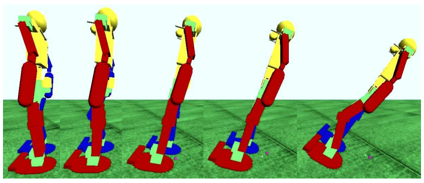
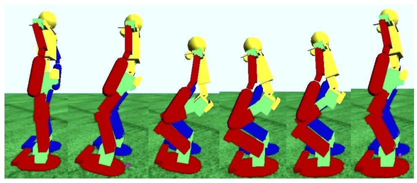
(a)  (b) 

Fig. 5. Robot disturbed by an impulse force on its chest while walking backward at speed $(0, -0.3, 0)$. (a) Robot falling when no stabilizing reflex is activated. (b) The stop walk reflex has been activated. The robot stops walking, crouches down and thereby prevents the fall.

Since in soccer games physical contact between the robots is unavoidable, the walking patterns are disturbed and the robots might fall. To prevent falls, we learn a model of the trunk attitude depending on the gait phase and the gait target vector. During walking, we aggregate the deviations from this model to an instability indicator. Protective reflexes, like slowing down and stopping in a stable posture (see Fig. 5), are triggered depending on the instability indicator [4]. If the disturbance is too large, and the fall cannot be prevented, the robot relaxes its joints in order to minimize mechanical stress at impact.

I. Lift the trunk and bring the forearms under the shoulders.
II. Move the COM projection as close as possible to the leading edges of the feet by bending in the spine, the hip pitch and the knee joints.
III. Straighten the arms to let the robot tip over the leading edges of the feet.
IV. Bring the body into an upright posture.

(a) → I → (b) → II → (c) → III → (d) → IV → (e)

Fig. 6. Standing up from the prone posture.

After falling, the robot's center of mass (COM) projection to the ground is outside the convex hull spanned by the foot-contact points. Additional support points, like knees, elbows, and hands, must be used in order to move the COM back inside the foot polygon. Using their attitude sensors, the robots detect a fall, classify the prone or supine posture and trigger the corresponding getting-up sequence. We designed the getting-up sequences in the simulator using sinusoidal trajectories [6]. Fig. 6 illustrates the four phases of getting up from the prone posture. The getting-up sequences work very reliably. Under normal circumstances, i.e. appropriate battery voltage, the routines worked with 100 successes in 100 tests.

6.4 Goalie Motions

The goalkeeper is capable of diving into both directions or to bend forward with spread arms. Fig. 7 shows Franz diving to the left. First, it moves its COM and turns its upper body towards the left while shortening the legs. As soon as it tips over its left foot, it starts straightening its body again. While doing so it is sliding on its hands and elbows. The fully extended robot covers the entire goal half. After the dive Franz gets up again, as described above.

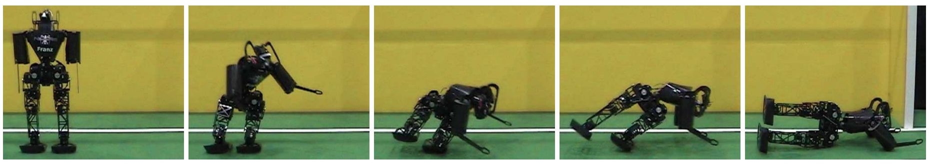


Fig. 7. Diving motion of the goalkeeper.

6.5 Soccer Skills

We used omnidirectional walking to implement basic soccer skills, like approaching the ball and dribbling. In addition to walking, we implemented kicking, obstacle avoidance, and defensive behaviors. Fig. 8 shows different phases of a kick.

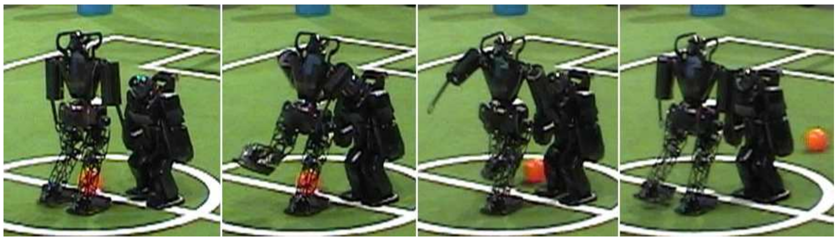


Fig. 8. Kicking the ball.

According to the current game situation, behaviors like searching the ball, positioning behind the ball, or avoiding obstacles are activated. These behaviors are implemented on the player level and use the actuator interface that basic skills of the lower layer provide. For example, they set the gait target vector or trigger a kick. Fig. 9 illustrates the inhibitory structure of the soccer behaviors and the actuator interface to the basic skills which reside on the lower layer. Further details on the soccer behaviors can be found in [**?**].

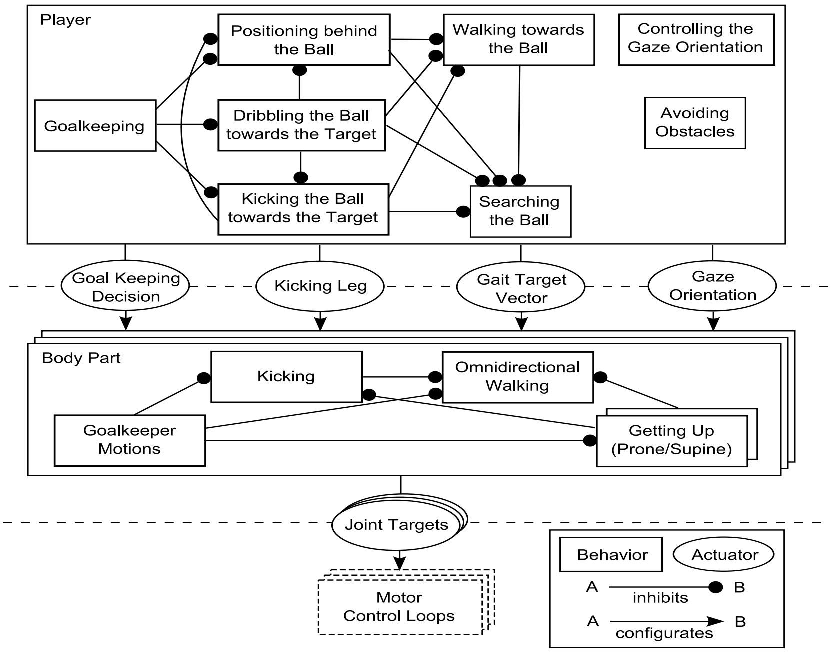


Fig. 9. Actuators, behaviors, and mutual inhibitions within the behavioral joint – body part – player hierarchy. Upper layer behaviors can configurate lower layer behaviors by manipulating the upper layer actuators.

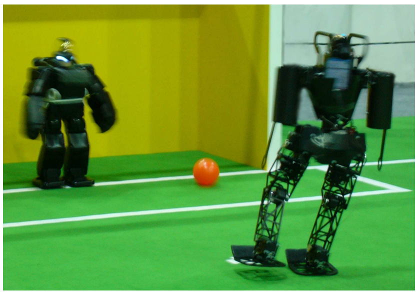
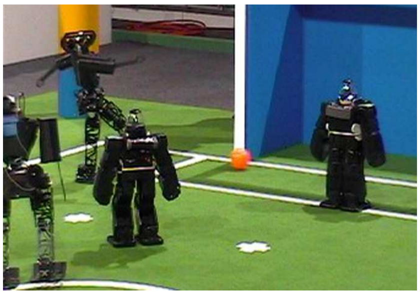
(a)  (b) 

Fig. 10. RoboCup 2006 Humanoid League final games: (a) Penalty Kick (NimbRo vs. Team Osaka); (b) 2 vs. 2 Soccer (NimbRo vs. Team Osaka).

6.6 Team Play

The importance of team behaviors is still low in the Humanoid League, as only two players per team compete. In our team, the players share perceptions via wireless communication. The ball perceptions communicated by other players are used for search. For the soccer play with two field players, we implemented simple but effective role negotiation between the players. As soon as one of our players has control of the ball, the other player goes to a defensive position between the ball and the own goal. Because we already coordinate two field players, adding a goalie would allow for 3 vs. 3 demonstration games without additional effort.

7 Conclusion

At the time of writing, May 20th, 2007, we made good progress in preparation for the competition in Atlanta. In addition to the KidSize 2006 robots, we are constructing a new generation of KidSize robots. We will play test games to select the best robots for RoboCup 2007.

The most recent information about our team (including videos) can be found on our web pages www.NimbRo.net.

Acknowledgements

Funding for the project is provided by Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under grant BE 2556/2-1,2,4 and Albert-Ludwigs-Universität Freiburg.

Fig. 11. Team NimbRo at RoboCup German Open 2007. From left to right: Hauke Strasdat, Martin Böhnert, and Konrad Meier. Back: Michael Schreiber, Jörg Stückler, Sven Behnke, and Hannes Schulz. Robots: Gerd, Franz, Robotinho, Paul, and Sepp.

Team Members

Currently, the NimbRo soccer team has the following members:

– Team leader: Dr. Sven Behnke
– Staff: Michael Schreiber, Dr. Maren Bennewitz
– Students: Martin Böhnert, Felix Faber, Konrad Meier, Hannes Schulz, and Jörg Stückler

References

1. Sven Behnke. Online trajectory generation for omnidirectional biped walking. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'06), Orlando, Florida*, pages 1597–1603, 2006.
2. Sven Behnke, Jürgen Müller, and Michael Schreiber. Using handheld computers to control humanoid robots. In *Proceedings of 1st International Conference on Dextrous Autonomous Robots and Humanoids (darh2005), Yverdon-les-Bains, Switzerland*, 2005.
3. Sven Behnke and Raul Rojas. A hierarchy of reactive behaviors handles complexity. In *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, pages 125–136. Springer, 2001.
4. Reimund Renner and Sven Behnke. Instability detection and fall avoidance for a humanoid using attitude sensors and reflexes. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China*, pages 2967–2973, 2006.
5. Russel Smith. Open Dynamics Engine. http://opende.sourceforge.net.
6. Jörg Stückler, Johannes Schwenk, and Sven Behnke. Getting back on two feet: Reliable standing-up routines for a humanoid robot. In *Proc. of The 9th International Conference on Intelligent Autonomous Systems (IAS-9), Tokyo, Japan*, pages 676–685, 2006.