NimbRo KidSize 2006 Team Description

Sven Behnke, Michael Schreiber, Jörg Stückler,
Hauke Strasdat, and Maren Bennewitz

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

Abstract. This document describes the RoboCup Humanoid League KidSize team NimbRo of Albert-Ludwigs-University Freiburg, Germany, as required by the qualification procedure for the competition to be held in Bremen in June 2006.

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 was established at Albert-Ludwigs-University Freiburg, Germany, in 2004. Our KidSize team participated with success at last year's RoboCup Humanoid League competition in Osaka, Japan. We scored the most goals (12/15) in the Penalty Kick round robin and reached the final in the 2 vs. 2 soccer games. This exciting game was won 2:1 by the titleholder, Team Osaka. We also came in second in the Technical Challenge. This resulted in the second place in the overall Best Humanoid ranking.

For the 2006 competition, we prepare not only for the Penalty Kick, the Technical Challenge, and the 2 vs. 2 soccer games, 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 2006 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. Sections 5 and 6 explain the generation of soccer behaviors in a hierarchy of agents and time-scales and the infrastructure needed to support a team of soccer playing robots, respectively.

2 Mechanical Design

Fig. 1 shows our humanoid robots Jupp and Sepp playing soccer. They are based on their predecessor Toni [2]. As can be seen, the robots have human-like

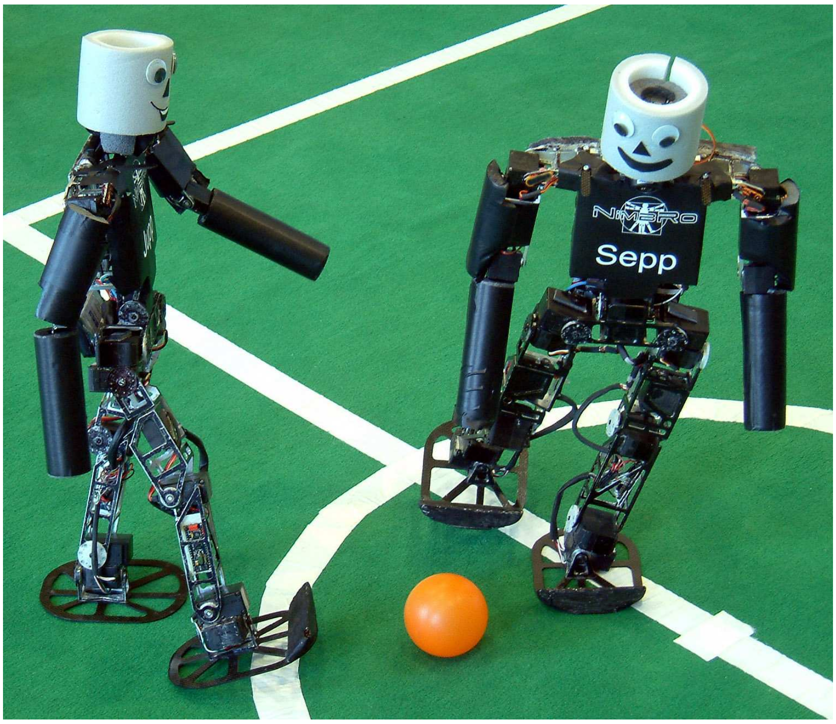

Fig. 1. NimbRo KidSize robots Jupp and Sepp playing soccer.

proportions. Their mechanical design focused simplicity, robustness, and weight reduction. The robots have a height of 60cm and a weight of only 2.3kg, including batteries.

Each robot is driven by 19 servo motors: 6 per leg, 3 in each arm, and one in the trunk. The six leg-servos allow for flexible leg movements. Three orthogonal servos constitute the 3DOF hip joint. Two orthogonal servos form the 2DOF ankle joint. One servo drives the knee joint.

We selected the S9152 servos from Futaba to drive the roll and yaw joints of the hips, the knees, and the ankles. These digital servos are rated for a torque of 200Ncm and have a weight of only 85g. The hip yaw joints need less torque. They are powered by DS 8811 servos (190Ncm, 66g). We augmented all servos by adding a ball bearing on their back, opposite to the driven axis. This made a stiff hinge joint construction possible. The arms do not need to be as strong as the legs. They are powered by SES640 servos (64Ncm, 28g). Two orthogonal servos constitute the shoulder joint and one servo drives the elbow joint.

The skeleton of the robots is constructed from aluminum extrusions with rectangular tube cross section. In order to reduce weight, we removed all material not necessary for stability. The feet and the forearms are made from sheets of carbon composite material. The elasticity of the feet and the carpet, the robots walk on, helps to maintain non-degenerate foot-ground contact, even when the supporting foot is not parallel to the ground. The heads of the robots are made of lightweight foam.

3 Electronics

Jupp and Sepp are fully autonomous. They are powered by high-current Lithium-polymer rechargeable batteries, which are located in their lower back. Two Kokam 2000H cells per robot last for about 30 minutes of operation. They can be discharged with 30A and have a weight of only 110g.

The servos of a robot are interfaced to three tiny ChipS12 microcontroller boards. One of these boards is located in each shank and one board is hidden in the chest. These boards feature the Motorola MC9S12C32 chip, a 16-bit controller belonging to the popular HCS12 family. We clock it with 24MHz. It has 2kB RAM, 32kB flash, a RS232 serial interface, CAN bus, 8 timers, 5 PWM channels, and 8 A/D converters. We use the timer module to generate pulses of 1...2ms duration at a rate of 180Hz in hardware. These pulses encode the target positions for the servos. Up to eight servos can be controlled with one board. In order to keep track of the actual servo movements, we interfaced their potentiometers to the A/D converters of the HCS12. By analyzing the temporal fine structure of these signals, we estimate not only the current servo positions, but also the PWM duty cycles of their motors.

In addition to these joint sensors, each robot is equipped with an attitude sensor and a compass. The attitude sensor is located in the trunk. It consists of a dual-axis accelerometer (ADXL203, $\pm$1.5g) and two gyroscopes (ADXRS 150/300, $\pm$150/300 deg/s). The four analog sensor signals are digitized with A/D converters of the HCS12 and are preprocessed by the microcontroller. The compass module is located in the head of the robot. It is interfaced to the timer module of the HCS12. Using pulse-width modulation, it indicates the robot's heading direction, relative to the earth's magnetic field.

The microcontrollers communicate with each other via a CAN bus at 1MBaud and with a main computer via a RS232 serial line at 115KBaud. Every 12ms, target positions for the servos are sent from the main computer to the HCS12 boards, which generate intermediate targets at 180Hz. This yields smooth joint movements. It is also possible to relax the digital servos. The microcontrollers send the preprocessed sensor readings back. This allows keeping track of the robot's state in the main computer. We use a Pocket PC as main computer [3], which is located in Jupp's chest. The FSC Pocket Loox 720 has a weight of only 170g, including the battery. It features a 520MHz XScale processor PXA-272, 128MB RAM, 64MB flash memory, a touch-sensitive display with VGA resolution, Bluetooth, wireless LAN, a RS232 serial interface, and an integrated 1.3 MPixel camera.

This computer runs behavior control, computer vision, and wireless communication. It is equipped with a Lifeview FlyCAM CF 1.3M that has been fitted to an ultra-wide-angle lens, which is located at the position of the larynx.

4 Perception

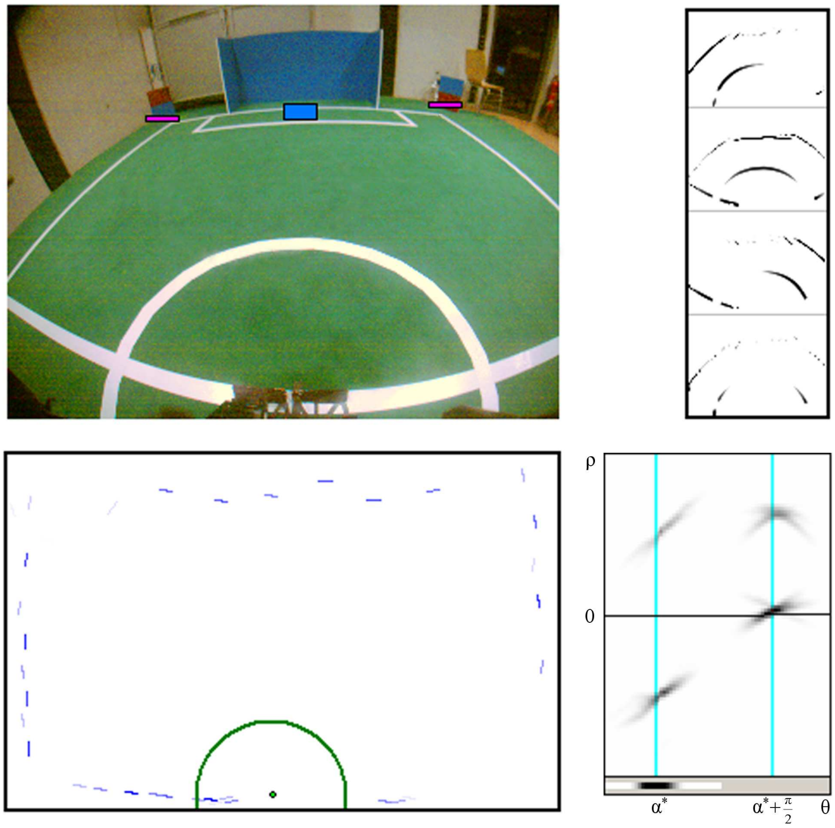

Fig. 2. Detection of field lines. An image captured by a walking robot is shown in the upper left. The responses of four orientated line detectors are shown in the upper right. They are used to detect oriented line segments, which are mapped into Cartesian egocentric coordinates (lower left). The center circle is detected and removed. The remaining line segments are mapped into Hough space (lower right), where the major orientation α^* is estimated and the main lines are detected.

Our robots need information about themselves and the situation on the soccer field to act successfully. We fuse the accelerometer and gyro readings to obtain an estimate of their attitude. We also estimate leg joint angles, and motor duties, and the heading direction.

The only source of information about the environment of our robots is their camera. The wide field of view of the CF-camera (about $112° \times 150°$) allows Jupp and Sepp to see at the same time their own feet and objects above the horizon. Figure 2 shows in the upper left part an image captured by a walking robot.

Our computer vision software detects the ball, the goals, the corner poles, the field lines, and other players based on their color. The remainder of Figure 2 illustrates line detection. We use four oriented line detectors (upper right) to detect points belonging to field lines. These points are aggregated to line segments. The detected lines segments are transformed into an egocentric Cartesian frame (lower left) by correcting for the lens distortion and the perspective camera projection. We detect the center circle and remove the line segments belonging to it. The remaining segments are transformed into the Hough space [7] (lower right) in order to estimate the major orientation α^* and to find the most significant field lines.

The observations of the field lines, the center circle, the goals, the corner poles, and the heading estimate from the compass are integrated in a particle filter [6] with the motion commands sent to the robot. This yields an estimate of the robots pose (x, y, θ) on the field, which is needed for team behaviors, such as kick-off.

5 Behavior Control

We control the robots using a framework that supports a hierarchy of reactive behaviors [5]. This framework allows for structured behavior engineering. Multiple layers that run on different time scales contain behaviors of different complexity. This 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 based on the readings of accelerometers and gyros. Abstract actuators give higher-level behaviors the possibility 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 Jupp and Sepp, 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. The lowest level of this hierarchy, the control loop within the servo, has been implemented by the servo manufacturer. It runs at about 300Hz for the digital servos. We monitor targets, actual positions, and motor duties.

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.

On the next higher level, we use this leg interface to implement omnidirectional walking [1]. 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. The maximal forward walking speed

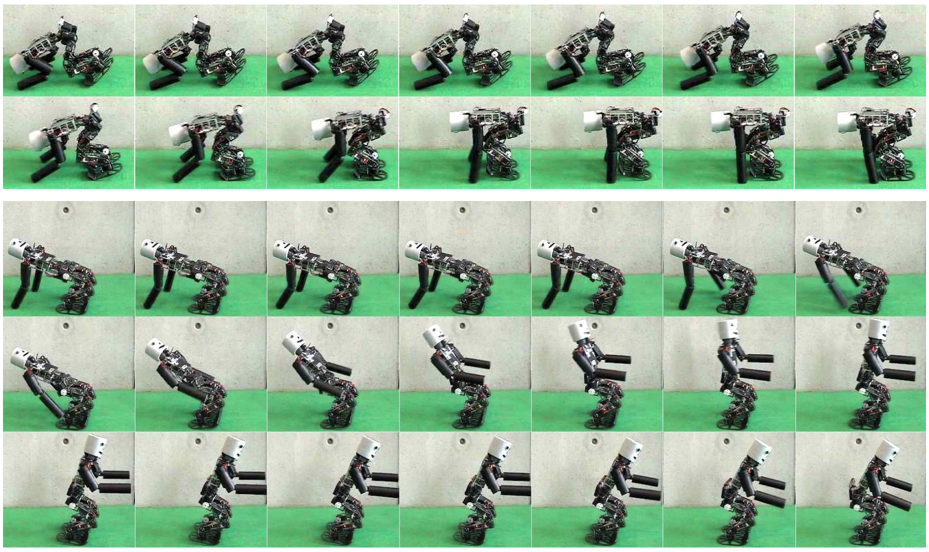

Fig. 3. Dynamic phase of getting up from the ground starting from the prone posture (upper sequence) and from the supine posture (lower sequence).

of the robots is approx. 25cm/s, but they walk slower in the vicinity of obstacles and the ball.

We used omnidirectional walking to implement some soccer skills, like approaching the ball and dribbling. In addition to walking, we implemented kicking, obstacle avoidance, and defensive behaviors. Because the direct physical contact of the robots during soccer games leads to falls, it was essential to implement getting-up as well [9]. Using their attitude sensors, the robots detect a fall, classify the prone or supine posture and trigger the corresponding getting-up sequence. Fig. 3 shows the dynamic part of both sequences. Currently, we are working on postural reflexes, which should minimize the number of falls.

6 Infrastructure

6.1 Communication

All our robots are equipped with wireless network adapters. We use the wireless communication to transmit debug information to an external computer, where it is logged and visualized. The wireless network is also used for transmitting the game state (kickoff, penalty ...) from the external PC to the robots.

The robots will also communicate with each other to negotiate roles and to share perceptions. The team play, we plan to implement, will be similar to the team behaviors that we tested at German Open 2005 with the RoboSapien robots [4].

6.2 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 is based on the Open Dynamics Engine [8].

7 Conclusion

At the time of writing, Feb 15th, 2006, we made good progress in preparation for the competition in Bremen. Currently, we are building a new generation of KidSize robots. We will play test games to select the best robots for RoboCup 2006.

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 and Albert-Ludwigs-Universität Freiburg.

Team Members

Currently, the NimbRo soccer team has the following members:

– Team leader: Dr. Sven Behnke
– Staff: Dr. Maren Bennewitz and Michael Schreiber
– Students: Konrad Meier, Reimund Renner, Alexander Schneider, Philip Sorst, Hauke Strasdat, and Jörg Stückler

References

1. Sven Behnke. Online trajectory generation for omnidirectional biped walking. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA'06), Orlando, Florida*, To appear 05/2006.
2. Sven Behnke, Jürgen Müller, and Michael Schreiber. Toni: A soccer playing humanoid robot. In *Proc. of 9th RoboCup Int. Symp., Osaka*, 2005.
3. 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.
4. Sven Behnke, Jürgen Müller, and Michael Schreiber. Playing soccer with RoboSapien. In *Proceedings of 2005 RoboCup International Symposium*, to appear July 2005.

5. 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.
6. Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.
7. Paul V.C. Hough. Machine analysis of bubble chamber pictures. In *Proc. of the Int. Conf. on High Energy Accelerators and Instrumentation (CERN)*, 1959.
8. Russel Smith. Open Dynamics Engine. http://opende.sourceforge.net.
9. 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*, To appear 03/2006.