NimbRo TeenSize 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 TeenSize 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 our TeenSize robot Robotinho. 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 TeenSize robot Max won the Penalty Kick at RoboCup 2005 in Osaka. Our TeenSize team participated with success at last year's RoboCup Humanoid League competition in Bremen, Germany. Our robots reached the Penalty Kick final. The final was won by Team Osaka. They also did well in the Race Walk, where they won against the team Pal Technology in the round robin. Our TeenSize robot Robotinho also attempted the dribbling challenge. It dribbled the ball around two poles before touching a pole. Overall, our TeenSize robots were the third best TeenSize team at RoboCup 2006.

For the 2007 competition, we prepare not only for penalty kick and the technical challenges, but also for 1 vs. 1 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. 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 Robotinho, our 2006 TeenSize robot. As can be seen, the robot has human-like proportions. Its mechanical design focused simplicity, robustness, and weight reduction.

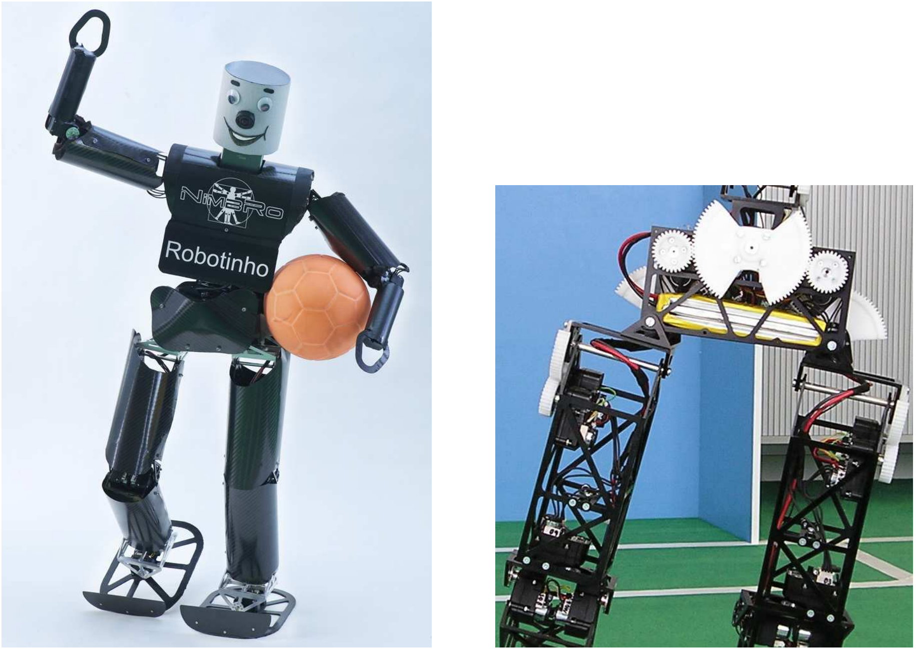

Fig. 1. NimbRo TeenSize 2006 robot Robotinho and close-up of its hip.

Robotinho is 100cm tall and has a total weight of about 5.2kg. Its 23 DOF are driven by a total of 35 Dynamixel DX-117 actuators: 11 per leg, 4 in each arm, and 5 in the trunk. Three orthogonal axes constitute the hip joint. Two orthogonal axes form the ankle joint. The knee joint has 1DOF. Robotinho has a very flexible trunk with 3DOF. Each arm consists of a 3DOF shoulder and an elbow joint.

All joints in the legs and the trunk, except for the yaw axes, are driven by two parallel actuators. The actuators are coupled in a master-slave configuration. This doubles the torque and lowers operating temperatures. The master-slave pair of actuators has the same interface as the single actuators used for all other joints. The hip and trunk yaw axes are reinforced by external 2:1 spur gears. The hip and trunk roll axes are reduced by 3:1, resulting in a holding torque of 222kg·cm at 16V.

Robotinho's skeleton is constructed from aluminum extrusions with rectangular tube cross section. In order to reduce weight, we removed all material not necessary for stability. Its feet and 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. Robotinho's head is made of lightweight foam. The robot is protected by a layer of foam and an outer shell of thin carbon composite material.

3 Electronics

Robotinho is fully autonomous. It is powered by high-current Lithium-polymer rechargeable batteries, which are located in its hip. Four Kokam 3200mAh cells last for about 30 minutes of operation.

The Dynamixel actuators have a RS-485 differential half-duplex interface. Robotinho is equipped with a CardS12 microcontroller board, which manages the detailed communication with all Dynamixels. These boards feature the Motorola MC9S12D64 chip, a 16-bit controller belonging to the popular HCS12 family. We clock it with 32MHz. It has 4kB RAM, 64kB flash, two serial interfaces, CAN bus, 8 timers, 8 PWM channels, and 16 A/D converters.

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, Robotinho is equipped with an attitude sensor, located in its trunk. It consists of a dual-axis accelerometer (ADXL203, ±1.5g) and two gyroscopes (ADXRS 300, ±300 °/s). The four analog sensor signals are digitized with A/D converters of the HCS12 and are preprocessed by the microcontroller. 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 microcontroller sends 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 [2], which is located in upper part of the robots. 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, and a RS-232 serial interface. 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. Robotinho's FlyCam lens also serves as nose. It looks downwards in forward direction.

4 Perception

Robotinho needs information about itself and the situation on the soccer field to act successfully. In this section, we detail proprioception, the visual perception of key objects and self-localization.

4.1 Proprioception

On the Pocket PC, the readings of accelerometers and gyros are fused to estimate the robot's tilt in roll and pitch direction. For each axis, the gyro bias is calibrated, assuming that over intervals of 2.4s the integrated bias-corrected gyro rates equal the difference between the tilts estimated from the accelerometers. Here we assume that, in the long run, the accelerometers measure the decomposition of the gravity vector. Combining the low-frequency components of the tilt estimated from accelerometers with the integrated bias-corrected gyro rates yields an estimate of the robot's attitude that is insensitive to short linear accelerations. As described above, joint angles, speeds, loads, temperatures, and voltages are also available. The temperatures and voltages are used to notify the user in case of overheating or low batteries.

4.2 Visual Object Detection

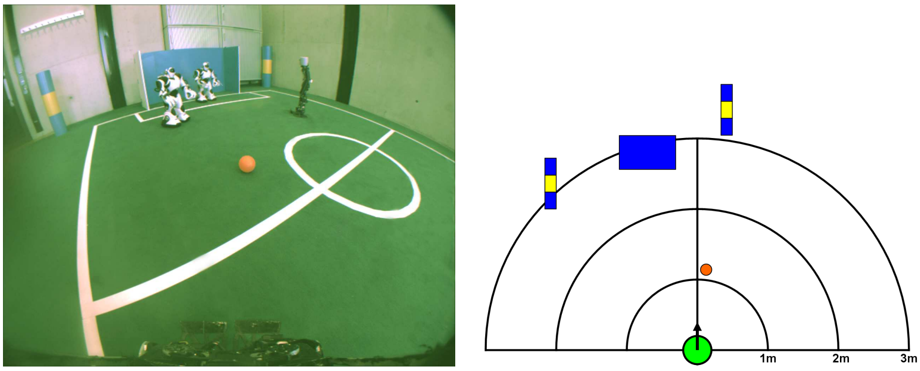

Fig. 2. Left: View onto the soccer field captured from the ultra-wide-angle FlyCam. Right: Egocentric coordinates of key objects (ball, goal, corner poles) detected in the image.

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°×150°) allows Robotinho to see at the same time its own feet and objects above the horizon. Figure 2 shows on the left an image captured from the perspective of a robot on the field.

Our computer vision software detects the ball, the goals, the corner poles, the field lines, and other players based on their color. It captures RGB images with a resolution of 320×240 pixels. The images are converted into the YUV color space to decrease the influence of different lighting conditions. The colors of individual pixels are classified with the pie-slice method [9]. We correct for the average brightness and for the darkening of the lens towards the periphery. 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 right part of Fig. 2.

These relative coordinates suffice for many relative behaviors, like positioning behind the ball while facing the goal. To implement more complex behaviors, we need the robot coordinates in an allocentric frame ((x, y)-position on the field and orientation θ).

4.3 Self-Localization

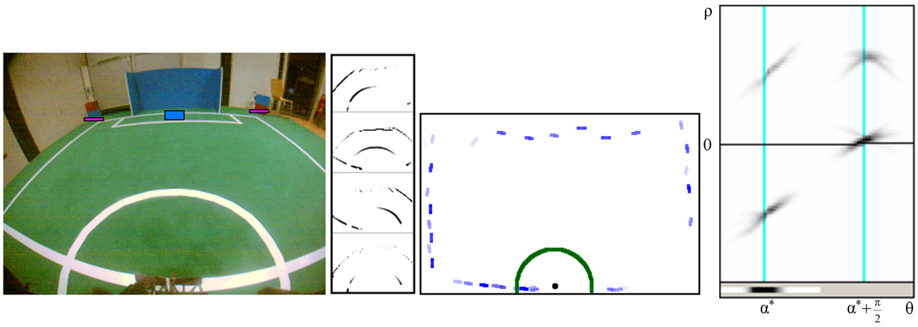

Fig. 3. Detection of field lines (left to right): image captured by a walking robot, responses of four orientated line detectors, oriented line segments with removed center circle, Hough space with major orientation α^* and main lines.

In addition to the already mentioned detected objects, the field lines are used for self-localization. Figure 3 illustrates the detection of the center circle and the field lines. We use four oriented line detectors to detect points belonging to field lines. The detectors make sure that green is present on both sides of the line. The detected line points are aggregated locally to larger line segments and transformed into an egocentric Cartesian frame.

Next, we locate the center circle [4]. The individual line segments vote for positions at a distance of the radius of the center circle, orthogonal to the orientation of the segment. By determining the largest cluster of points, we find and eliminate the segments corresponding to the center circle. The remaining line segments are transformed into the Hough space [6]. Since we have already estimated the orientation of the line segments, we only have to vote for a small subset of orientation bins. Utilizing the property that the field lines are orthogonal, we determine the main orientation α^* (modulo $\frac{\pi}{2}$) and find for each corresponding orientation the two most significant lines.

The observations of the field lines, the center circle, the goals, and the corner poles are integrated in a particle filter [5] with the motion commands sent to the robot. We compare the observed positions of landmarks with the expected positions of the landmarks that should be visible from the particle poses. For the field lines, we use not only the Hough parameters (θ, ρ) to assess similarity, but also two parameters that describe line position and length. We also apply a motion model that moves the particles according to the motion commands sent to the robot. The particle filter yields an estimate of the robots pose (x, y, θ) on the field. More details can be found in [8].

5 Behavior Control

We control Robotinho 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 Robotinho, we use three levels of this hierarchy: individual joint – body part – entire robot. 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 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 a kinematic interface for the body parts. It allows, for example, to independently change leg extension η, leg angle $\theta_{\mathbf{Leg}}$, and foot angle $\theta_{\mathbf{Foot}}$, as illustrated in Fig. 4. A detailed description of the kinematic leg interface is given in [1]. Several basic skills, described below, use this kinematic interface. The next higher level contains soccer behaviors which are executed at a rate of 41.7Hz.

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 2.44Hz for Robotinho. The three basic walking directions can be smoothly combined. Robotinho is able to walk in every direction and to change its 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 Robotinho is approx. 20cm/s, but he walks slower in the vicinity of obstacles and the ball. We used omnidirectional walking to implement basic soccer skills,

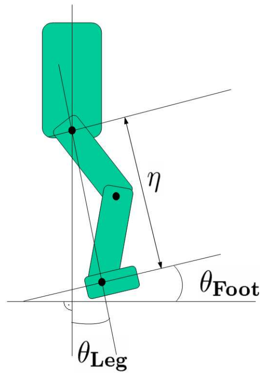

Fig. 4. Kinematic interface to a leg.

like approaching the ball and dribbling. In addition to walking, we implemented kicking. Fig. 5 shows Robotinho kicking during the Final at RoboCup 2006 and for a demo at the Humanoids 2006 conference in Genoa.

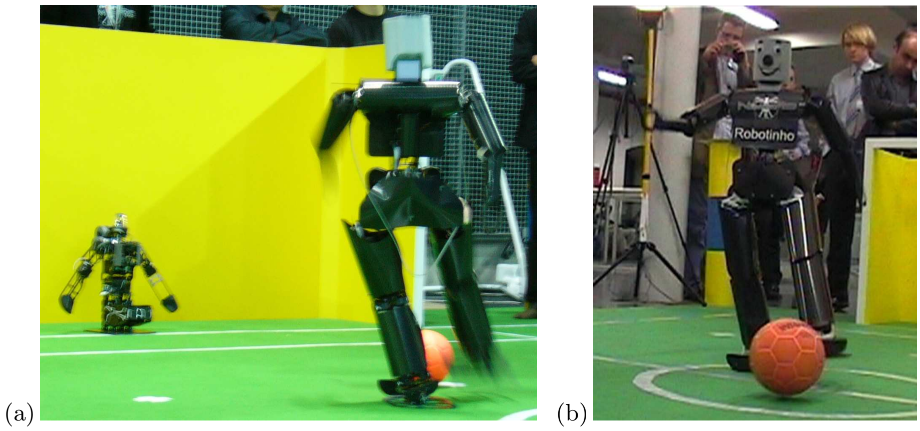

Fig. 5. Robotinho kicking: (a) RoboCup 2006 TeenSize Penalty Kick final vs. Team Osaka; (b) Demo at the Humanoids 2006 conference.

6 Infrastructure

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

• **Wireless Communication:** Robotinho's Pocket PC is equipped with a wireless network adapter. 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 (e.g. start/stop) from the external PC to the robot.

• **Simulation:** In order to be able to design behaviors without access to the real hardware, we implemented a physics-based simulation for Robotinho. This simulation uses the Open Dynamics Engine [7].

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 TeenSize 2006 robot Robotinho, a new TeenSize robot is under construction. We will select the best robot for RoboCup 2007.

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

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

Fig. 6. 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.

Acknowledgements

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

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. F. de Jong, J. Caarls, R. Bartelds, and P.P. Jonker. A two-tiered approach to self-localization. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: Robot Soccer World Cup V*, volume 2377 of *Lecture Notes in Artificial Intelligence*, pages 405–410. Springer Verlag, Berlin, 2002.
5. Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo localization for mobile robots. In *Proceeding of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.
6. Paul V.C. Hough. Machine analysis of bubble chamber pictures. In *Proc. of the Int. Conf. on High Energy Accelerators and Instrumentation (CERN)*, 1959.
7. Russel Smith. Open Dynamics Engine. http://opende.sourceforge.net.
8. Hauke Strasdat, Maren Bennewitz, and Sven Behnke. Multi-cue localization for soccer playing humanoid robots. In *Proceedings of 10th RoboCup International Symposium, Bremen, Germany*, 06/2006.
9. Peter J. Thomas, Russel J. Stonier, and Peter J. Wolfs. Robustness of colour detection for robot soccer. In *Proceedings of 7th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, volume 3, pages 1245–1249, 2002.