DESIGNING A TEAM OF SOCCER-PLAYING HUMANOID ROBOTS

Sven Behnke, Michael Schreiber, Maren Bennewitz, Jörg Stückler, Hauke Strasdat, and Johannes Schwenk
University of Freiburg, Computer Science Institute
Germany

Abstract

Robotic soccer superseded chess as a challenge problem and benchmark for artificial intelligence research and poses many challenges for robotics. The international RoboCup championships grew to the most important robotic competition worldwide. After preliminary competitions, for the first time soccer games with humanoid robots were played in Osaka 2005. This paper describes the mechanical and electrical design of our robots, which took part as team NimbRo at the competitions. The paper also covers the software used for perception, behavior control, communication, and simulation. Our robots performed well at RoboCup 2005. They came in second and third in the overall Best Humanoid ranking, next only to the titleholder, Team Osaka.

Speaker: Sven Behnke, University of Freiburg, Germany, behnke@informatik.uni-freiburg.de
Topic: Humanoids, Soccer Robots
Keywords: humanoid robots, robotic soccer, RoboCup

1 Introduction

What drives thousands of researchers worldwide to devote their creativity and energy to make robots bring a ball into a goal? The answer lies not only in the fascination of the soccer game, but rather in the quest to advance the fields of artificial intelligence research and robotics.

AI researchers started to investigate games early-on. Already in the Fifties of the last century, Simon predicted that computers would be able to win against the human world champion within ten years [11]. Playing chess was viewed as epitome of intelligence. The dominant view at that time was that human intelligence could be simulated by manipulating symbols. While the world champion in chess was defeated by a machine in 1997 [9], human intelligence is still far from being understood.

The basis for intelligent action is the perception of the world. Already this seemingly easy task frequently exceeds the capabilities of current computer systems. Perceptual processes, which interpret the flood of stimuli that stream into our senses and make it accessible for behavior control, are mostly unconscious. Hence, we are not aware of the difficulties involved. The performance of our perceptual system becomes clear only when trying to solve the same task with machines. This applies to behavior control as well. Human locomotion, for example, does not seem to be problematic. That walking and running on two legs is not an easy task becomes clear only when one tries to implement it on a real robot.

Based on these observations, a view on intelligence has established itself over the last two decades that does not rely on manipulating symbols, but emphasizes the interaction of an agent with its environment [4, 10]. The term embodiment stresses the importance of having a body as the physical basis for intelligence. Situatedness of an agent in a rich environment enables feedback from the actions of the agent to sensory signals. The complexity of the interaction between an agent and its environment is increased significantly when the environment does not only contain passive objects, but other agents as well.

1.1 RoboCup Competitions

Motivated by the successes in the chess domain, the RoboCup Federation organizes since 1997 international robotic soccer competitions. Similar competitions are organized by the competing FIRA. The long-term goal of the RoboCup Federation is to develop by the year 2050 a team of humanoid soccer robots that wins against the FIFA world champion [8]. The soccer game was selected for the competitions, because, as opposed to chess, multiple players of one team must cooperate in a dynamic environment. Sensory signals must be interpreted in real-time and must be transformed into appropriate actions. The soccer competitions do not test isolated components, but two systems compete with each other. The number of goals

Figure 1: Some of the robots that competed at RoboCup 2005 in the Humanoid League.

scored is an objective performance measure that allows comparing systems that implement a large variety of approaches to perception, behavior control, and robot construction. The presence of opponent teams, which continuously improve their system, makes the problem harder every year. Such a challenge problem focuses the effort of many research groups worldwide and facilitates the exchange of ideas.

The RoboCup championships grew to the most important robotic competition worldwide. In the last RoboCup, which took place in July 2005 in Osaka, Japan, 330 teams from 31 countries competed. The total number of participants was about 2.000 and 182.000 spectators watched the competitions. Likewise, the media coverage was enormous. In addition to the soccer competitions, since 2001, competitions for the search of victims of natural disasters and the coordination of rescue forces are held (RoboCupRescue). Furthermore, there are competitions for young researchers (RoboCupJunior).

1.2 RoboCupSoccer

The soccer competitions at RoboCup are held in five leagues. Since the beginning, there is a league for simulated agents, a league for small wheeled robots which are observed by cameras above the field (SmallSize), and a league for larger wheeled robots where external sensors are not permitted (MiddleSize). A league for the Sony Aibo dogs was added in 1999 (Four-legged) and a league for humanoid robots was established in 2002.

Different research issues are addressed in the different leagues. In the simulation league, team play and learning are most advanced. In the wheeled robot leagues, the robot construction (omnidirectional drives, ball manipulation devices), the perception of the situation on the field (omnidirectional vision systems, distance sensors), and the implementation of basic soccer skills (approaching, controlling, dribbling, and passing the ball) are still in the center of the activities. Because the robot hardware is fixed in the Four-legged League, the participating teams focus on perception and behavior control. Here, the control of the 18 degrees of freedom (DOF) poses considerable challenges.

As the performance of the robots increases, the competition rules are made more demanding by decreasing the deviations from the FIFA laws. This permanently increases the complexity of the problem. It can also be observed that solutions like team play, which have been developed in leagues abstracting from real-world problems, are adopted in hardware leagues, as the basic problems of robot construction, perception, locomotion, and ball manipulation are solved better.

1.3 Humanoid Soccer Robots

In the Humanoid League, robots with a human-like body plan compete with each other. The robots must have two legs, two arms, a head, and a trunk. Size restrictions make sure that the center of mass of the robots is not too low, that the feet are not too large, and so on. The robots are grouped in two size classes: KidSize (up to 60cm) and TeenSize (65cm-130cm).

The humanoid robots must be able to walk on two legs. While in the first years of the league, it was allowed to remotely control the robots and to use a power cable, since 2004, the robots must be fully autonomous. The robots may communicate with each other via a wireless network, but help from outside the field is not permitted, neither by humans nor by computers.

Because the construction and the control of humanoid robots is significantly more complex than that of wheeled robots, initially, there were only preliminary competitions held, but no soccer games played, in the Humanoid League. The robots had to footrace around a pole and faced each other in penalty kicks. Since 2005, soccer games take place. At the German Open in April 2005, two teams of autonomous RoboSapien robots (Brainstormers und NimbRo) showed demonstration games. In July 2005, at the RoboCup in Osaka, 2 vs. 2 soccer games were played in the KidSize class.

The Humanoid League rules have been derived from the FIFA laws. Some simplifications apply, however. For example, the offside rule is not observed. Key objects are color-coded in order to simplify the perception of the game situation. The playing field is green with white lines, the goals are painted blue and yellow, the ball is orange, and the robots are mostly black. The two teams are marked with magenta and cyan patches, respectively.

The remainder of this paper 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

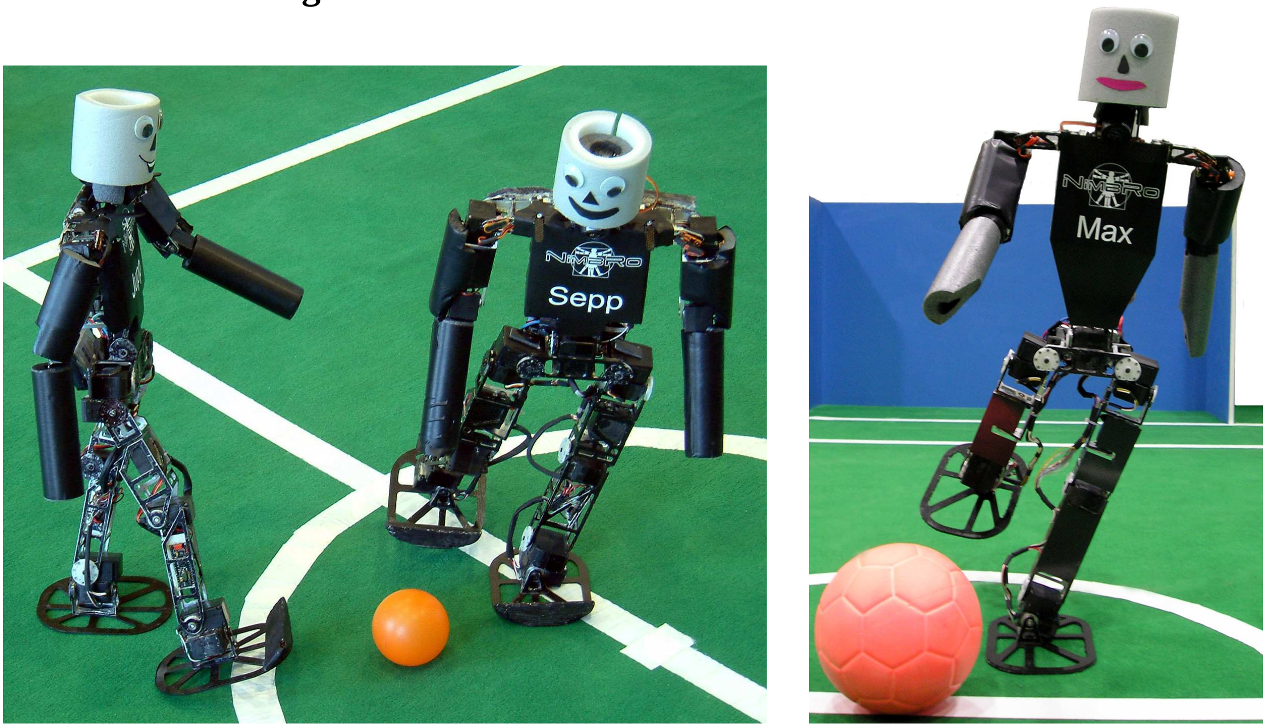


Figure 2: NimbRo 2005 KidSize robots Jupp and Sepp (left) and TeenSize robot Max (right) playing soccer.

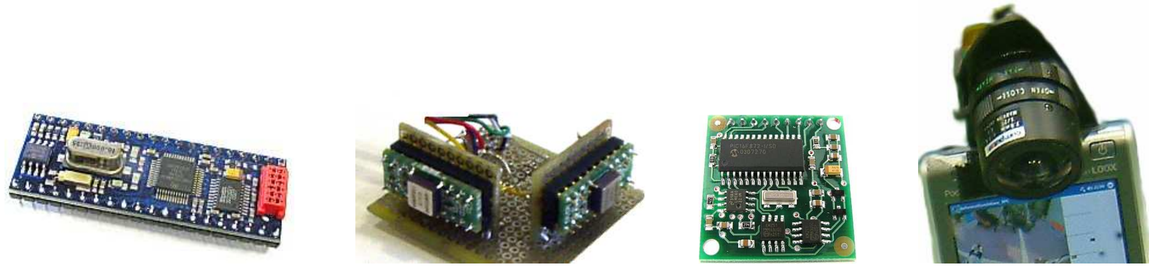
Fig. 2 shows on the left our KidSize robots Jupp and Sepp playing soccer and on the right our TeenSize robot Max, ready to kick. These robots are based on their predecessor Toni [2]. As can be seen, the robots have human-like proportions. Their mechanical design focused simplicity, robustness, and weight reduction. The KidSize robots have a height of 60cm and a weight of only 2.3kg, including batteries. Max is scaled to 75cm and weighs 2.4kg.

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). The trunk pitch joint is also driven by a S9152. 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



(a) (b) (c) (d)

Figure 3: Electronic components used: (a) ChipS12 microcontroller board; (b) attitude sensor; (c) compass; (d) Pocket PC with ultra-wide-angle CF-camera.

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 (see Fig. 3(a)). 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, shown in Fig. 3(b), 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 $^\circ$/s). The four analog sensor signals are digitized with A/D converters of the HCS12 and are preprocessed by the microcontroller. The compass module (see Fig. 3(c)), 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, which is located in the chest. The FSC Pocket Loox 720, shown in Fig. 3(d), 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. The lens surfaces at the position of the larynx and looks downwards.

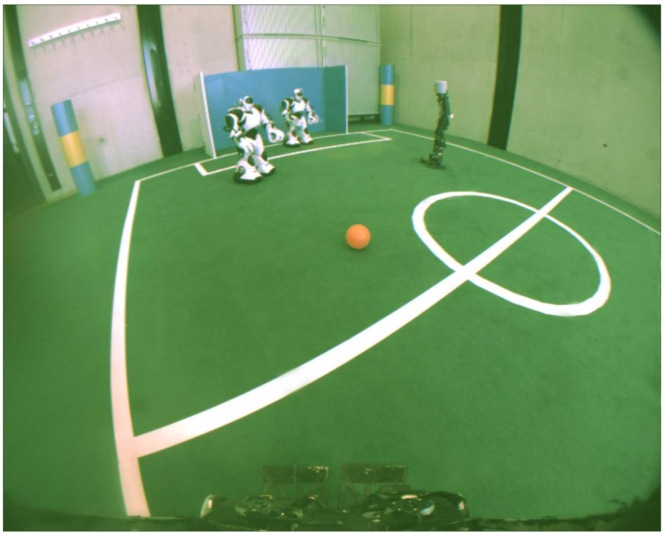
4 Perception

Our robots need information about themselves 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. We also estimate leg joint angles, motor duties, and the heading direction (from the compass).

4.2 Visual Object Detection



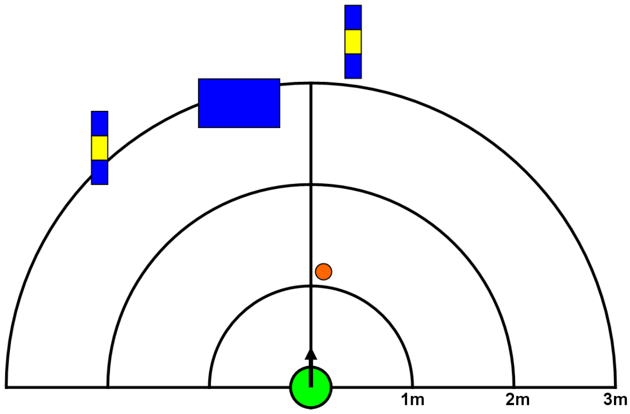


Figure 4: Left: View onto the soccer field captured from a KidSize robot. Right: Egocentric coordinates of key objects (ball, goal, corner poles) detected in the image.

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 them to see at the same time their own feet and objects above the horizon. Figure 4 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 [15]. 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. 4.

These relative coordinates suffice for many relative behaviors, like positioning behind the ball while facing the goal. To implement team behaviors, such as kick-off, we need the robot coordinates in an allocentric frame ((x, y)-position on the field and orientation θ). In addition to the already mentioned detected objects, the field lines are used for self-localization.

4.3 Self-Localization

Figure 5 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. This discards spurious responses of the line detectors and allows estimating the orientations of the line segments better. The detected line segments are transformed into an egocentric Cartesian frame by correcting for the lens distortion and the perspective camera projection.

Before we apply the Hough transform to find the best fitting lines for the extracted oriented segments, we locate the center circle. Whenever large parts of the circle are visible in the image, this can impair the line detection. Parts of the circle can be misclassified as short lines and the circle can potentially affect the estimation of the main orientations. To avoid these problems, we first identify the center circle following the approach presented by de Jong et al. [5]. We consider the individual line segments and vote for the 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 and identifying the segments that voted for it, we find the segments

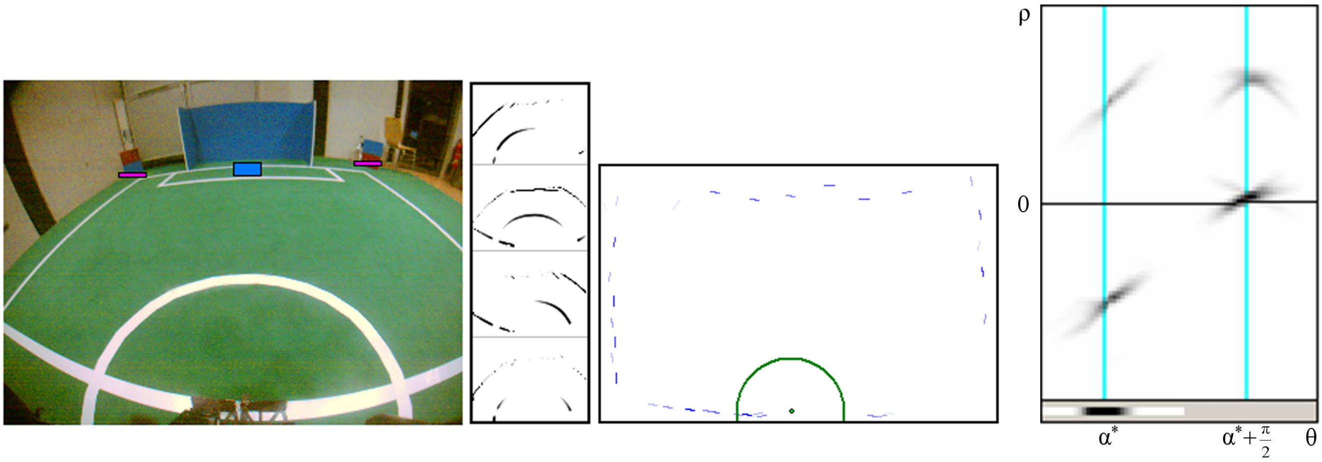


Figure 5: Detection of field lines. An image captured by a walking robot is shown on the left. The next part shows responses of four orientated line detectors. They are used to detect oriented line segments, which are mapped into Cartesian egocentric coordinates. The center circle is detected and removed. The remaining line segments are mapped into Hough space (right), where the major orientation α^* is estimated and the main lines are detected.

corresponding to the center circle. To avoid false positive detections, we only interpret a cluster as the center circle if the line segments that voted for it have a large range of orientations. The corresponding segments are eliminated.

The remaining line segments are transformed into the Hough space [7] in order to estimate the major orientation α^* and to find the most significant field lines. The Hough transform is a robust method to find lines fitting a set of 2D points. It relies on a transformation from the Cartesian plane in the Hough domain. The following curve in the Hough domain is associated with a point (x, y) in the Cartesian plane:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \tag{1}$$

Here, ρ is the perpendicular distance from the origin and θ is the angle with the normal. This curve describes all the lines that go through (x, y), since each point in the Hough space corresponds to a line in the 2D Cartesian space.

Usually, each detected line segment (x, y) votes for all bins $h(\theta, \rho)$ which fulfill Eq. (1). Since we have already estimated the orientation of the line segments, we only have to vote for a small subset of bins, which reduces the computational costs. In particular, we accumulate its likelihood in the bins $h(\theta \pm \epsilon, \rho)$ that correspond to the estimated orientation θ of a segment. Here, ϵ indicates that we consider a local neighborhood of θ whose bins are also incremented. In this way, we direct the search towards lines that fit the preestimated orientations. In our current implementation, we use a discretization of 2.5° and 6cm for the Hough space. In general, to locate lines in the Hough space one has to search the entire space for local maxima. In the RoboCup domain, we only have two possible orthogonal orientations for the field lines. This allows us to use a robust method for finding lines that additionally reduces the computational costs: We can determine the two main orientations by adding the bins corresponding to α and $\alpha + \frac{\pi}{2}$, with $\alpha \in [0; \frac{\pi}{2}[$ and finding the maximum:

$$\alpha^* = \underset{\alpha=(\theta_i \bmod \frac{\pi}{2})}{\operatorname{argmax}} \sum_{\rho_j} h(\theta_i, \rho_j) \tag{2}$$

Finally, we search for maxima in the bins of α^* and $\alpha^* + \frac{\pi}{2}$, respectively. In this manner, we extract from the Hough space four field lines, two for each main orientation.

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. 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 [14].

5 Behavior Control

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. 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 computed from the readings of accelerometers and

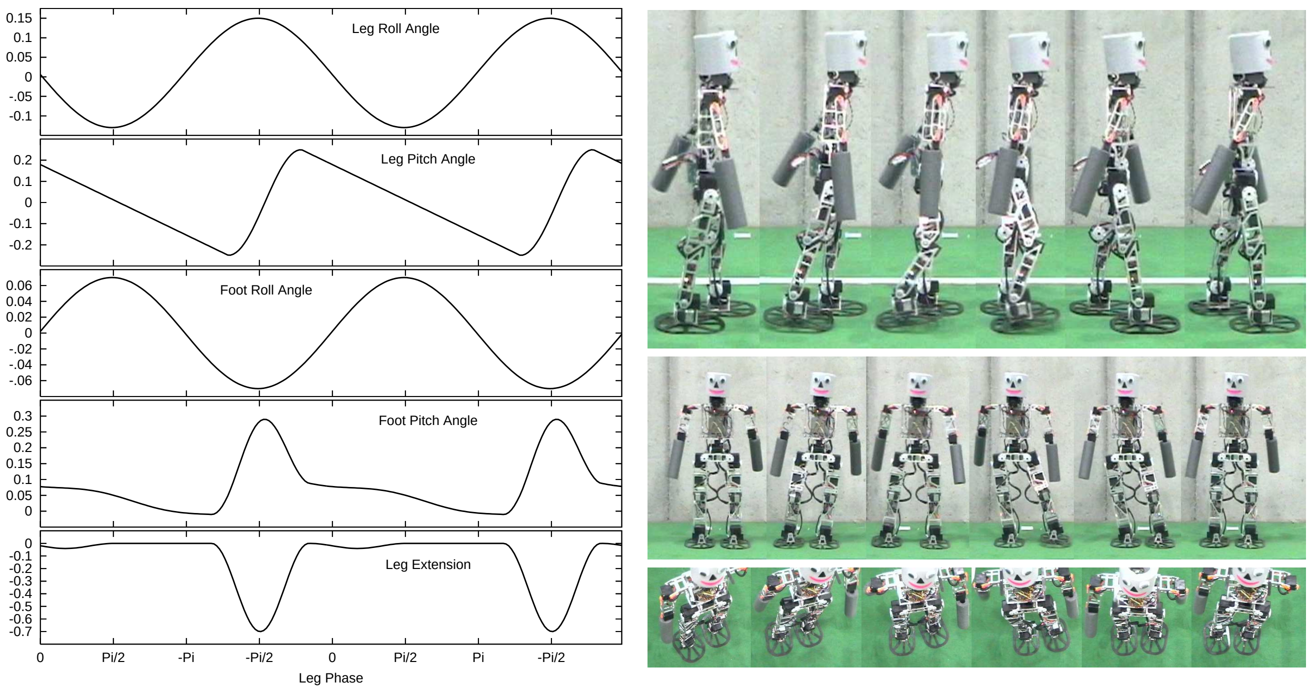


Figure 6: Left: Trajectories for forward walking of robot Jupp. Right: Walking to the front, to the side, and turning.

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]. 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. The left part of Fig. 6 shows the trajectories generated for forward walking. Walking to the side and rotating on the spot is generated in a similar way. The right side of the figure shows image sequences of our robot Jupp walking into the three basic directions. These 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. 15cm/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.

Since in soccer games physical contact between the robots is unavoidable, the walking patterns are disturbed and the robots might fall. Hence, they must be able to detect the fall, to recognize their posture on the ground, and to get back into an upright 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 [13]. Fig. 7 illustrates the four phases of getting up from the prone posture. The dynamic phase III is shown in Fig. 8. Fig. 9 and Fig. 10 illustrate getting up from the supine 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. They were also crucial in the 2 vs. 2 games, because the robots could continue play after a fall.

Phase I. Lift the trunk and bring the forearms under the shoulders.

Phase 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.

Phase III. Straighten the arms to let the robot tip over the leading edges of the feet.

Phase IV. Bring the body into an upright posture.

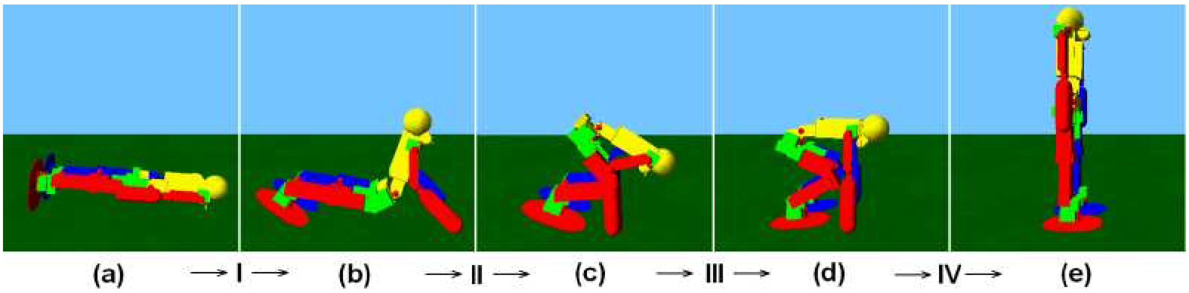


Figure 7: (a)-(e) Starting and end positions of phases I-IV when standing up from the prone posture.

Figure 8: Dynamic phase III of getting up from the ground starting from the prone posture.

Phase I. Move the upper body into a sit-up posture and move the arms into a supporting position behind the back.

Phase II. Move into a bridge-like position using the arms as support.

Phase III. Move the COM over the feet by swinging the upper body to the front.

Phase IV. Move the body into an upright posture.

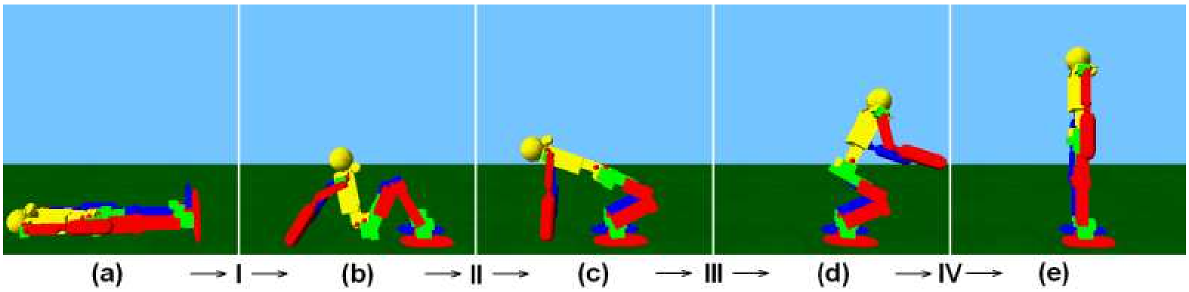


Figure 9: (a)-(e) Starting and end positions of phases I-IV when standing up from the supine posture.

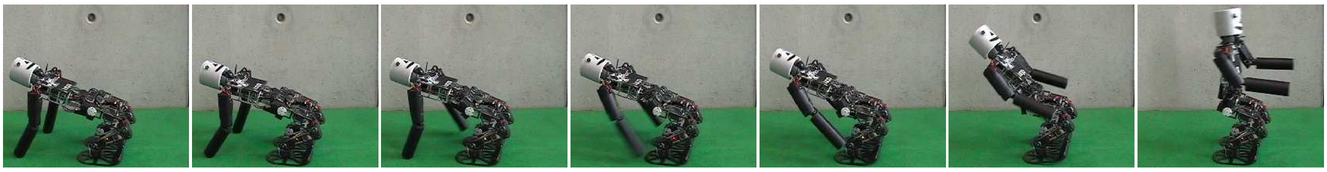


Figure 10: Dynamic phase III of getting up from the ground starting from the supine posture.

6 Infrastructure

In addition to the robots themselves, some infrastructure components are needed to support a team of soccer playing robots. They include wireless communication and a simulator.

6.1 Wireless Communication

The Pocket PCs of our robots are equipped with wireless network adapters. We use the wireless communication to transmit via UDP debug information to an external computer, 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 ...) from the external PC to the robots. The robots communicate with each other to share perceptions. For example, if one robot does not see the ball, it might use the ball observation of its teammate to find the ball again.

The importance of team behaviors is still low in the Humanoid League, as only 2 players per team have competed so far. In Osaka 2005, most teams assigned one player to keep the goal clear and used the other player as field player. For 3 vs. 3 demonstration games, which are scheduled for the RoboCup in Bremen 2006, we plan to implement some role negotiation between the players. This team behavior will be similar to the team behaviors that we tested at German Open 2005 with the RoboSapien robots. For example, it will prevent multiple robots from taking the initiative to approach the ball.

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 [12]. It was very helpful for the design of getting-up behaviors, goalie behaviors, and team behaviors. The simulator is also used to develop learning algorithms, which are then applied to the real robot.

7 Results

Our robots performed well at RoboCup 2005, where 20 teams from nine countries competed in the Humanoid League. Two photos from the competition are shown in Fig. 11.

In the technical challenge, where the robots had to walk slalom around three poles and to kick the ball against a fourth pole, we came in second (Jupp) and third (Max). Jupp and Sepp scored the second highest number of goals in the KidSize penalty kick competition. Max won the penalty kick competition in the TeenSize class 3:0 against Aria (Iran). In the KidSize class, we reached the final in the 2 vs. 2 soccer games, against the titleholder Team Osaka [16]. The exciting game ended 2:1 for Osaka. In the overall Best Humanoid ranking, we came in second (KidSize) and third (TeenSize), next only to Team Osaka. Videos showing the omnidirectional walking and the getting up of our robots, as well as their performance at RoboCup 2005 can be found at http://www.NimbRo.net/media.html.

8 Conclusions

This paper described the mechanical and electrical design of our robots, which successfully took part as team NimbRo at the RoboCup 2005 competitions. We detailed the software used for perception, behavior control, communication, and

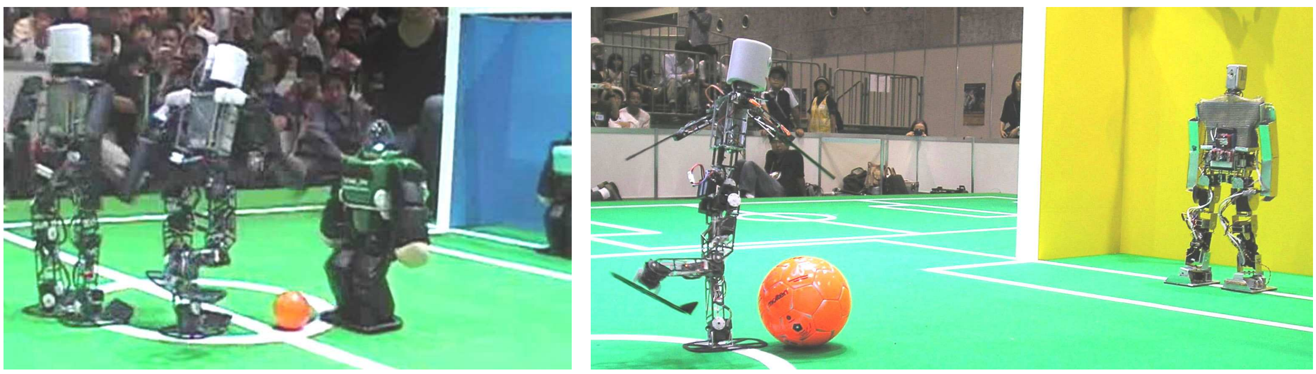


Figure 11: RoboCup 2005. Left: 2 vs. 2 KidSize final (Jupp and Sepp of team NimbRo vs. Team Osaka). Right: TeenSize penalty kick (Max of team NimbRo vs. Aria).

simulation.

Playing soccer with humanoid robots is a complex task, and the development has only started. So far, there has been significant progress in the Humanoid League, which moved in its few years from remotely controlled robots to soccer games with fully autonomous humanoids. Indeed, the Humanoid League is currently the most dynamic RoboCupSoccer league. We expect to see the rapid progress continue as more teams join the league. Many research issues, however, must be resolved before the humanoid robots reach the level of play shown in other RoboCupSoccer leagues. For example, the humanoid robots must maintain their balance, even when disturbed. Currently, we are working on postural reflexes, which should minimize the number of falls.

In the next years the speed of walking must be increased significantly. We work on automatic gait optimization to increase both speed and stability. At higher speeds, running will become necessary. We recently started to explore this direction. The visual perception of the soccer world must become more robust against changes in lighting and other interferences. We continuously improve our computer vision software to make it more reliable.

Among the biggest challenges when designing a team of soccer playing robots is the integration of subsystems. While it is not that hard to develop a vision system or to implement walking, it is not easy to operate these components simultaneously within a humanoid robot. The weight and power consumption of the components plays a role that should not be underestimated. High reliability of all parts, as well as the handling of exceptions are indispensable in order to survive a game without breakdowns. As the performance of the system is not determined by the strongest component, but by the weakest link in the chain, this component deserves our attention in future research.

Acknowledgements

Funding for the project is provided by Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under grant BE 2556/2-1 and by 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*, To appear 05/2006.

[2] Sven Behnke, Jürgen Müller, and Michael Schreiber. Toni: A soccer playing humanoid robot. In *Proceedings of 9th RoboCup Int. Symp., Osaka*, 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] Rodney A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.

[5] 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.

[6] 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.

[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] Hiroaki Kitano and Minoru Asada. The RoboCup Humanoid Challenge as the millennium challenge for advanced robotics. *Advanced Robotics*, 13(8):723–737, 2000.

[9] Monty Newborn. *Kasparov versus Deep Blue: Computer chess comes of age*. Springer, 1997.

[10] Rolf Pfeifer and Christian Scheier. *Understanding Intelligence*. The MIT Press, Cambridge MA, 1999.

[11] Herbert A. Simon and Alan Newell. Heuristic problem solving: The next advance in operations research. *Operations Research*, 6(1):1–10, 1958.

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

[13] 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.

[14] Hauke Strasdat, Maren Bennewitz, and Sven Behnke. Multi-cue localization for soccer playing humanoid robots. In *Proceedings of 10th RoboCup International Symposium, Bremen, Germany*, to appear 06/2006.

[15] 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.

[16] N. Yamato, Y. Akazawa, H. Ishiguro, T. Takahashi, T. Maeda, T. Imagawa, H. Takayama, N. Mitsunaga, and T. Miyashita. VisiON Nexta: Fully autonomous humanoid robot. In *RoboCup 2005 Humanoid League Team Descriptions, Osaka, Japan*, 2005.

About the Authors

The team NimbRo at the RoboCup 2005 competition in Osaka. From left to right: Front row: robots Max, Jupp, and Sepp; Middle row: Michael Schreiber, Hauke Strasdat, and Jörg Stückler; Back row: Johannes Schwenk, Maren Bennewitz, and Sven Behnke.

Dr. Sven Behnke received his MS CS (Dipl.-Inform.) from Martin-Luther-Universität Halle-Wittenberg in 1997. Afterwards, he moved to Freie Universität Berlin, where he received his PhD in Computer Science in 2002. During 2003, he worked as a postdoc at the International Computer Science Institute in Berkeley, CA. Since 2004, he has been heading the junior research group Humanoid Robots at Albert-Ludwigs-Universität Freiburg.

Michael Schreiber studies computer science at Freie Universität Berlin. Since 2004, he constructs humanoid robots at Albert-Ludwigs-Universität Freiburg.

Dr. Maren Bennewitz received her MS CS (Dipl.-Inform.) from Rheinische Friedrichs-Wilhelm-Universität Bonn in 1999. Afterwards, she moved to Albert-Ludwigs-Universität Freiburg, where she received her PhD in 2004. She joined the Humanoid Robots group after her graduation.

Jörg Stückler studies computer science at Albert-Ludwigs-Universität Freiburg. He developed the simulator, getting-up behaviors, and goalie behaviors.

Hauke Strasdat received his BA CS from Universität Osnabrück in 2005. Afterwards, he moved to Freiburg to continue studying towards a MS CS. He developed computer vision and self-localization. Since March 2006, he is visiting the CS department of University of Washington, Seattle.

Johannes Schwenk studies computer science at Albert-Ludwigs-Universität Freiburg. He developed getting-up behaviors and goalie behaviors.