# Hierarchical Reactive Control for a Team of Humanoid Soccer Robots

Sven Behnke, Jörg Stückler, Michael Schreiber, Hannes Schulz, Martin Böhnert, and Konrad Meier
Computer Science Institute
University of Freiburg, Germany
{ behnke | stueckle | schreibe | schulzha | boehnert | meierk }@informatik.uni-freiburg.de

*Abstract*— **Humanoid soccer serves as benchmark problem for artificial intelligence research and robotics. Every year, more teams are competing, e.g., in the RoboCup Humanoid league. As the robots manage the basic skills of walking, kicking, and getting up better, teams can focus on soccer skills and team coordination. The complexity of soccer behaviors and team play calls for structured behavior engineering.**

**In this paper, we describe the design of the behavior control software for the Humanoid League team NimbRo. The control software is based on a framework that supports a hierarchy of reactive behaviors. It is structured both as an agent hierarchy (joint – body part – player – team) and as a time hierarchy. The speed of sensors, behaviors, and actuators decreases when moving up in the hierarchy. The lowest levels of this framework contain position control of individual joints and kinematic interfaces for body parts. At the next level, basic skills are implemented. These are used by soccer behaviors like searching for the ball, approaching the ball, avoiding obstacles, and defending the goal. Finally, on the tactical level, the robots communicate via a wireless network to negotiate roles and use allocentric information to configure the soccer behaviors.**

**Our robots won all humanoid soccer competitions of RoboCup 2007, which took place in Atlanta, GA.**

## I. Introduction

Humanoid soccer serves as benchmark problem for artificial intelligence research and robotics. The RoboCup Federation and FIRA organize international robotic soccer competitions. The long-term goal of RoboCup is to develop by the year 2050 a team of humanoid soccer robots that wins against the FIFA world champion [1]. 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 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 2007 in Atlanta, GA, 321 teams from 39 countries competed. The total number of participants was about two thousand.

Fig. 1. Some of the humanoid robots that competed at RoboCup 2007.

In the RoboCup Humanoid League, fully autonomous 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 into two size classes: KidSize (up to 60cm) and TeenSize (80cm-160cm).

The RoboCup Humanoid League was established in 2002 and has developed quickly since. It is now the league with the largest number of participants. 29 teams from 14 countries competed in the Humanoid League. Some of the participating robots are shown in Fig.1. As the humanoid soccer robots manage the basic skills of walking, kicking, and getting up better, the research teams start to focus on soccer skills and on the coordination of the individual players.

Playing soccer is not a trivial task. The ball might be at any position on the field and the robots need to search for it if they have lost track of its position. The robots must also perceive at least the two goals and the other players. Higher-level behaviors require self-localization on the field. As two robots play together, there is need for coordination. While some teams use one dedicated goalie and one field player, other teams use two field players. This makes dynamic role assignment necessary. Last, but not least, in soccer games robots of the two teams interact physically when going for the ball. This disturbs walking and leads to falls. The robots need to get up from the ground by themselves in order to continue play. As a result of these difficulties, only a fraction of the participating teams were able to play decent soccer games.

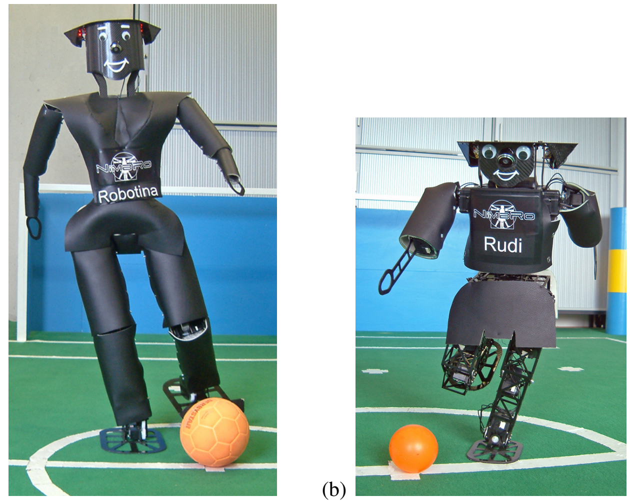
To implement the behavior control software for the humanoid soccer robots of our team NimbRo, we used a framework that supports a hierarchy of reactive behaviors [2]. This framework has been originally developed for the FU-Fighters SmallSize robots. It was later adapted to the FU-Fighters MiddleSize robots and also used by CMU in the Four-Legged League [3]. We adapted it for the control of soccer playing humanoid robots by extending the agent-hierarchy to: joint – body part – player – team. The lowest levels of this hierarchy contain position control of individual joints and kinematic interfaces for body parts. At the next level, basic skills like omnidirectional walking, kicking, and getting-up behaviors are implemented. These are used at the player level by soccer behaviors like searching for the ball, approaching the ball, avoiding obstacles, and defending the goal. Finally, on the tactical level, the robots communicate via a wireless network to negotiate roles and use allocentric information to configure the soccer behaviors.

The remainder of this paper is organized as follows. After reviewing some of the related work, we describe the mechanical and electrical design of our KidSize and TeenSize 2007 robots in Sec. III. Proprioception and visual perception of the game situation are detailed in Sec. IV. Sec. V describes our behavior control framework. The implementation of basic skills is covered in Sec. VI. The design of our soccer behaviors and their use on the allocentric tactical level are described in Sec. VII and Sec. VIII, respectively. Finally, we present the results of using the proposed system at RoboCup 2007.

## II. Related Work

The other RoboCupSoccer leagues have been facing the complexity of soccer games for some years now. There, tools for structured behavior engineering have been developed. For example, Jaeger and Christaller proposed the Dual Dynamics architecture [4], which has been used in the MiddleSize League. The architecture distinguishes elementary behaviors, which implement a target dynamics, and complex behaviors, which control the activation of elementary behaviors. The Dual Dynamics approach has been combined with a planner in the DD&P framework [5]. The DD-Designer also supports team coordination [6].

Another tool used in the MiddleSize League is the BAP-framework of Utz et al. [7], which allows for specifying hierarchical, event-driven, behavior-based control systems. In the Four-Legged League, the German Team developed XABSL [8]. It allows for XML-based specification of hierarchies of behavior modules that contain state machines for decision making. State transitions are modeled as decision trees. Parts of the German Team system are used now in the Humanoid League by Darmstadt Dribblers, Humanoid Team Humboldt, and BreDoBrothers. Another example for a behavior architecture used in more than one league is the architecture proposed by Laue and Röfer [9], which combines action selection and potential field motion planning. It was used to control SmallSize and Aibo soccer robots.




(a) (b)

Fig. 2. NimbRo 2007: (a) TeenSize robot Robotina; (b) KidSize robot Rudi.

Our architecture for hierarchical reactive behavior control builds on Dual Dynamics, but extends it to multiple levels in the time hierarchy and organizes behaviors in an agent hierarchy. This modular approach facilitates the development and maintenance of behaviors. Flexible inhibition mechanisms allow, e.g., for smooth transitions between behaviors or for hysteresis of behavior activation. Multiple behaviors are activated concurrently in the different levels and agents of the hierarchy. Even within a behavior layer, multiple behaviors can be active simultaneously.

## III. NimbRo 2007 Robots

We constructed five new robots for RoboCup 2007: Rudi, Jürgen and Lothar play in the KidSize class. Bodo is the TeenSize goalie and Robotina is the TeenSize penalty kick striker. Fig. 2 shows Rudi and Robotina. As can be seen, the robots have human-like proportions. Their mechanical design focused on simplicity, robustness, and weight reduction.

### A. Main Computer and Camera System

As compared to the NimbRo 2006 robots, which were controlled by a Pocket PC, the 2007 robots have a much stronger main computer and a high-bandwidth vision system that has a 360° field-of-view. The new robots are controlled by a tiny PC, a Sony Vaio UX, which features an Intel 1.33GHz ULV Core Solo Processor, 1GB RAM, 32GB SSD, a 4.5” WSVGA touch-sensitive display, 802.11a/b/g WLAN, and a USB2.0 interface. The weight of the UX is only 486g. Three IDS uEye UI-1226LE industrial USB2.0 cameras provide omnidirectional sight. The cameras feature a 1/3” WVGA CMOS sensor, global shutter, and are equipped with ultra-wide angle lenses. Each camera has a $90° \times 140°$ field-of-view. In order to cover all directions, the cameras are heading to the front (0°) and to the left/right rear (±120°).

### B. Mechanical Design

The NimbRo 2007 robots have also stronger actuators, compared to the NimbRo 2006 robots. The KidSize robots

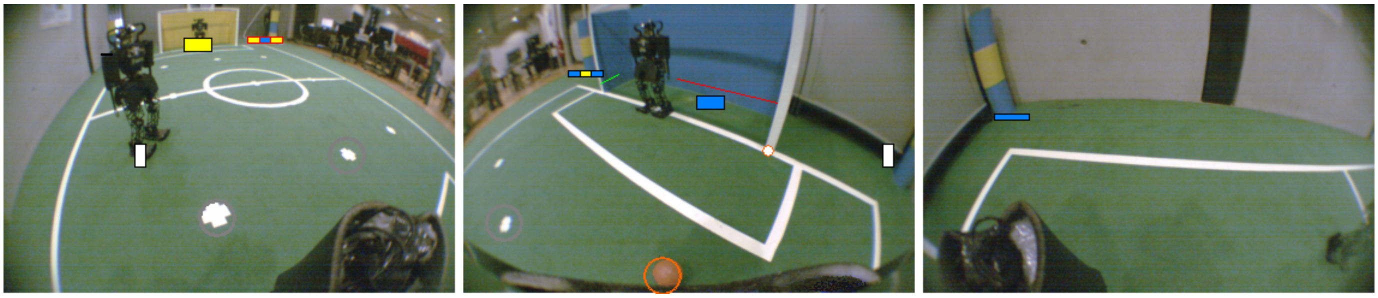


Fig. 3. Images captured by the three cameras. The detected objects are marked: goals (blue and yellow rectangle), ball (orange circle), field makers (gray circles), corner poles (horizontal blue and yellow lines), goal post (white circle), and obstacles (white vertical rectangles). The free goal area is marked on the left and the right of the goalie with a green and red line, respectively.

Rudi, Jürgen and Lothar are 60cm tall and have a total weight of 4kg. The robot Bodo, used mainly as TeenSize goalie, has been extended to 83cm. These four robots are driven by 20 Dynamixel actuators: 6 per leg, 3 in each arm, and two in the trunk. For all leg joints, except hip yaw, and for the trunk pitch joint, we use large RX-64 actuators (116g, 64kg·cm). All other joints are driven by smaller DX-117 actuators (66g, 37kg·cm).

The TeenSize robot Robotina is 122cm tall and has a total weight of about 8.75kg. Its 21 DOF are driven by a total of 33 Dynamixel actuators. The additional joint is the roll axis in the trunk. 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. Dynamixel RX-64 actuators are used in the legs and DX-117 actuators are used in the trunk and in the arms. The ankle, hip, and trunk yaw/roll axes are reinforced by external 2:1/3:1 spur gears, respectively, resulting in a holding torque of 384kg·cm (39Nm) in the ankle and hip roll joints. The knee is not reduced with an external spur gear, because it needs to move quickly. Instead, a torsional spring is added in parallel to the knee actuators. This spring supports stretching the knee. It is designed to compensate for the weight of the robot when it is standing with partially bent knees.

The skeleton of all five 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, the forearms, and the robot heads are made from sheets of carbon composite material. The upper part of the smaller robots and the entire body of Robotina is protected by a layer of foam and an outer shell of synthetic leather.

### C. Electrical Design

Our soccer robots are fully autonomous. They are powered by high-current Lithium-polymer rechargeable batteries, which are located in their hip. Five Kokam 1250mAh cells are used for the KidSize robots. Robotina has five Kokam 3200mAh cells. The batteries last for about 25 minutes of operation.

The Dynamixel actuators have a RS-485 differential half-duplex interface. Each robot is equipped with a CardS12X microcontroller board, which manages the detailed communication with all Dynamixels. These boards feature the Motorola MC9S12XDP512 chip, a 16-bit controller belonging to the popular HCS12X family. The controller has an I/O coprocessor and many interfaces, including serial lines and 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, the robots are equipped with an attitude sensor, located in the trunk. It consists of a dual-axis accelerometer (ADXL203, $\pm$1.5g) and two gyroscopes (ADXRS, $\pm$300 °/s). The four analog sensor signals are digitized with A/D converters of the HCS12X 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.

## IV. Perception

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

### A. 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.

### B. Visual Object Detection

The only information sources for our robots about their environment are three cameras. Our computer vision software captures and interprets images with 752×480 pixels at an

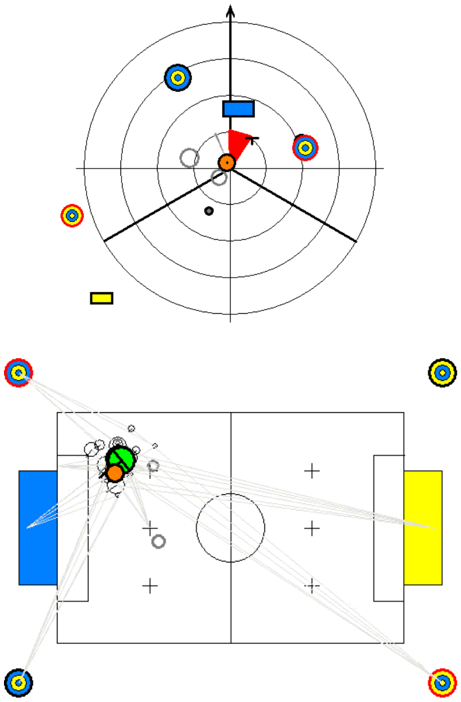


Fig. 4. Egocentric and allocentric representations of the game situation constructed by the computer vision module.

aggregated frame rate of about 32fps. The wide field of view of the cameras allows the robots to see objects close to their own feet and objects above the horizon in all directions at the same time. Fig. 3 shows three camera images that have been captured simultaneously with marked objects. 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. The computer vision software also detects the goal posts, the goal area not covered by the goalie, and the white field markers. We estimate the coordinates of detected objects 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 part of Fig. 4. Here, the objects detected by the three 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.

### C. *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 the ball with other team members, we need the robot coordinates in an allocentric frame ($(x, y)$-position on the field and orientation $\theta$). We solve self-localization by triangulation over pairs of landmark observations, i.e. detected goals, goal posts, corner poles, and field markers. When observing more than two landmarks, the triangulation results are fused based on their confidence. We apply a mean-shift procedure to exclude outliers from the final pose estimation. Again, the results of self-localization are integrated over time and a motion model is applied. The lower part of Fig. 4 illustrates the resulting allocentric representation.

## V. Behavior Architecture

We control the robots using a framework that supports a hierarchy of reactive behaviors [2]. This framework allows for structured behavior engineering. Multiple layers that run on different time scales contain behaviors of different complexity. When moving up the hierarchy, the speed of sensors, behaviors, and actuators decreases. At the same time, they become more abstract.

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 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 speed, which configures the gait engine, described below, implemented in the lower control levels.

The behaviors within one layer of the behavior framework are activated according to the current state of its sensors. Activation is indicated by an activation factor in the interval $[0, 1]$. Each active behavior can manipulate the actuators in its layer. If multiple behaviors try to manipulate the same actuator, the actuator is set to the weighted sum of desired values, where the activation factors are used as weights. To prevent conflicting behaviors from being active at the same time, behaviors can inhibit other behaviors. If an inhibiting behavior is not completely active, the inhibited behaviors share the remaining activation, such that the activation factors sum to one.

The control hierarchy of our soccer robots is arranged in an agent hierarchy, where

- multiple joints (e.g. left knee) constitute a body part (e.g. left leg),
- multiple body parts constitute a player (e.g. field player), and
- multiple players constitute a team.

The behavior framework manages all but the motor control loop within the Dynamixel actuators, which has been implemented by Robotis.

## VI. Basic Skills

Fundamental for playing soccer are the abilities to walk and to kick. As body contact between the physical agents is unavoidable, the capability of getting up after a fall is also essential. To act as a goalkeeper, the robot must be able to

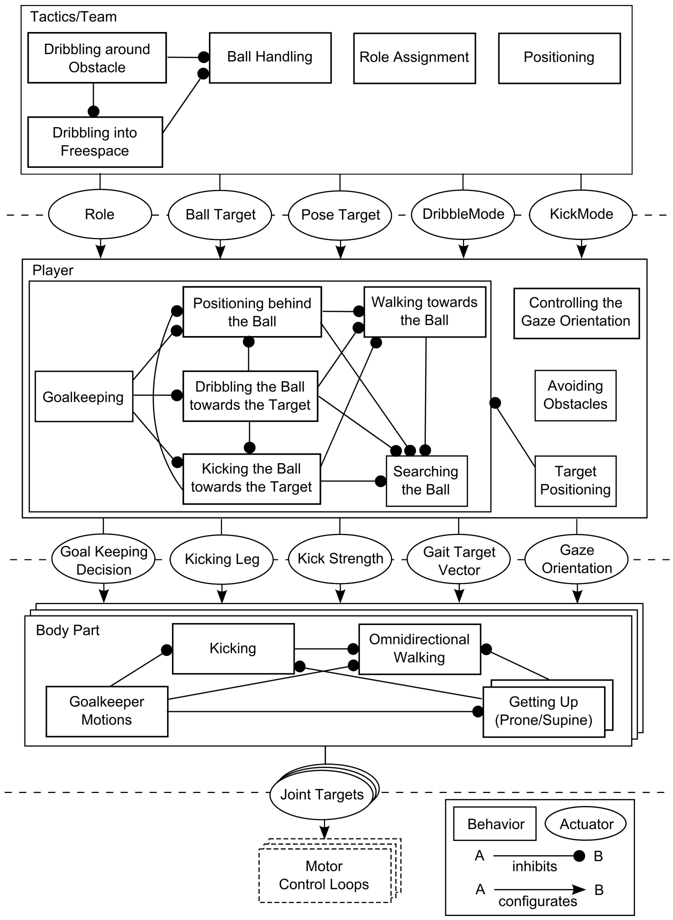



Fig. 5. Actuators, behaviors, and mutual inhibitions within the behavioral hierarchy. Upper layer behaviors can configure lower layer behaviors by manipulating the upper layer actuators. The resulting values of the actuators depend on the activation factors and the inhibitory structure of the manipulating behaviors.

perform special motions. These basic skills are implemented on the body part layer through behaviors which generate target positions for individual joints at a rate of $83.3Hz$. To abstract from the individual joints, we implemented here a kinematic interface for the body parts. An example for such an interface for the legs can be found in [10].

Some basic skills can be configured through actuators from the next higher level in our behavior control system. This makes abstraction from joint trajectory generation possible, when soccer specific tasks are implemented. Fig. 5 illustrates the inhibitory structure of the basic skills and the interface that they provide for the next behavior level.

In the following we will give a brief description of each basic skill. A more detailed description can be found in [11].

*1) Omnidirectional walking:* The ability to move into any direction, irrespectively of the orientation, and to control the rotational speed at the same time has advantages in many domains, including RoboCupSoccer. Omnidirectional drives are used by most teams in the wheeled leagues, and omnidirectional walking is heavily used in the Four-legged League. It is much easier to position robots for kicking and to outmaneuver opponents when using omnidirectional locomotion.

The omnidirectional gait of our humanoid soccer robots is achieved by generating walking patterns online [10]. Translational and rotational walking speeds can be set to values within continuous ranges for each individual direction. Besides speed limits, there are no restrictions to the combination of walking speeds. The gait target vector $(v_x, v_y, v_\theta)$ can be changed continuously while the robot is walking. The key ingredients of the omnidirectional gait are shifting the weight from one leg to the other, shortening of the leg not needed for support, and leg motion in walking direction. By controlling the foot angles in dependency of the angular velocity of the trunk, we were able to enhance the robustness and the speed range of the gait significantly. During walking it is also possible to change the twist of the trunk. Higher level behaviors can specify this rotation through a gaze orientation actuator.

*2) Kicking:* Our robots are able to perform a parameterizable kick motion to the front. An actuator allows behaviors in the upper level to trigger the kick with both, the left and the right leg. The kick strength is also configurable through an actuator. To correct for smaller angular misalignments to the kick target, the kick angle is adjusted by rotating the kicking leg in yaw direction.

*3) Getting up after a Fall:* Since in soccer games physical contact between the robots is unavoidable, the walking patterns are disturbed and the robots might fall. Using their attitude sensors, the robots detect a fall, relax their joints before impact, classify the prone or supine posture, and trigger the corresponding getting-up sequence. We designed the getting-up sequences in a physics-based simulator using sinusoidal trajectories [12]. The getting-up sequences work very reliably. Under normal circumstances, i.e. appropriate battery voltage, the routines worked with 100 successes in 100 tests.

*4) Goalkeeper Motions:* The goalkeeper is capable of diving into both directions or to bend forward with spread arms.

## VII. Soccer Behaviors

The next higher level of our behavior control framework contains soccer behaviors which are executed at a rate of 41.7Hz. They build on the basic skills and have been designed for 2 vs. 2 soccer games.

An essential part of this level is an adequate representation of the current game situation. The visual perception supplies relative distance, angle, and perceptual confidence for the ball, the own goal, the opponent goal, the center angle of the largest goal area not covered by the goalie, and the nearest obstacle. In the offensive role, the relative position and confidence of the largest free area in the opponent goal is used as the target to kick at (ball-target), if it is perceived confidently. Else the opponent goal is used as ball-target. To kick the ball, the robot has to position with its kicking leg behind the ball. Thus, the corresponding relative target position of the robot, denoted as behind-ball-position, and the kicking leg are contained in the game state. The decision for the kicking leg is made at every time step, depending on the relative position of the ball and the line from ball to ball-target. If the robot has to approach the ball-to-target-line from the right, it kicks with the left leg, and vice versa. To avoid oscillations, decisions are only made as long as the distance of the robot to the ball-to-target-line exceeds a threshold. When playing as defensive field player, the own goal is used as ball-target, such that the position behind the ball is set to a defensive position between ball and own goal. The components of the current game state are provided to the soccer behaviors through sensors. Fig. 6 illustrates the positioning of offensive and defensive field player with an example.

The robot also maintains hypotheses about the relative ball location that are used for searching the ball. If a kick is triggered, one hypothesis is set to the predicted end position of the ball, as due to limitations in visibility range the visual perception could fail to detect the ball. Additionally, hypotheses are maintained for the perceptions of the ball by other players on the field. The confidences in these hypotheses depend on the self-localization and ball perception confidences of the other players and the self-localization confidence of the robot itself. The relative positions of the hypotheses are altered according to the motion model. Their confidences are discounted by the time since the last update.

*1) Searching the Ball:* If the ball is not perceived, the robot has to explore the field for it. In the case that a confident ball hypothesis exists, it gazes and walks towards it. If no confident hypotheses exist, it first stops walking and sweeps its upper trunk over a range of $\pm\frac{\pi}{2}$, because motion blur reduces the visibility range of balls during walking. If still no ball is in sight, it walks to the farer goal, and then turns to the other goal. After this, it repeats the search process.

*2) Walking towards the Ball:* As it is possible to perceive the ball, while the opponent goal is occluded, the robot is kept close to the ball. The behavior is not activated, if the robot is close to the own goal.

*3) Positioning behind the Ball:* If ball and opponent goal are visible, the robot positions behind the ball, as given in the current game state. While the distance to the target position is large, the robot rotates towards the target position, such that it can approach it by mainly combining forward walking with turning. If it is near the target position, the robot aligns itself towards the ball-target. For intermediate distances, the gait rotation is interpolated linearly between both alignment targets. The behavior also handles the case when the ball is located between the robot and the behind-ball-position. Here, the robot walks around the ball by walking towards the target position but avoiding the ball-to-target-line. When playing as defensive field player, the robot rotates towards the ball at any distance without avoiding the ball-to-target-line.

*4) Kicking the Ball towards the Target:* This behavior is activated as soon as the behind-ball position has been reached with a certain precision in angle to the ball-target and in distance to the target position. The active behavior triggers a kick with the correct kicking leg. If the ball comes into a kicking position by chance, the behavior initiates a kick with the corresponding leg. As the robot has to come to a complete stop before the kicking motion can be executed, the robot can cancel the kick, if the ball moves away in the meantime.

*5) Dribbling the Ball towards the Target:* If positioning behind the ball was not successful for a longer time, or at kick-off, the robot dribbles the ball towards the ball-target for some time. Target-oriented dribbling is achieved by steering towards the ball, if it is still far away. If it is close, the robot aligns the ball in the center in front of its feet by lateral motion, and aligns its orientation towards the ball-target. The better this alignment is, the faster the robot moves in forward direction. At intermediate distances, the alignment targets are interpolated linearly. Dribbling inhibits kicking the ball.

*6) Avoiding Obstacles:* After a fall, the robot needs valuable time to get back on its feet. The main reason for our robots to fall is physical contact with other robots. Hence, obstacle avoidance is an important feature. The visual perception supplies the behavior with the nearest obstacle. If it is detected closely in front of the robot, obstacle avoidance is activated. The avoidance sets the gait target actuator to a constant and a variable part of the direction from obstacle to robot, which norm is proportional to the distance to the obstacle. If the ball is between obstacle and robot, the variable avoidance is weakened, such that the robot moves more aggressively behind the ball. A stuck situation is indicated by a resulting gait target vector that is small in length for a longer time. In this case, the robot may sidestep the obstacle, if the ball is not between the obstacle in the front and the robot and is perceived on one side of the obstacle. The action is canceled, if either the preconditions for sidestepping do not hold anymore or a certain amount of time has elapsed since sidestepping has been activated.

*7) Controlling the Gaze Orientation:* A gaze control behavior keeps the ball within an angular range of $\pm\frac{\pi}{4}$ in the front camera by rotating the upper trunk in yaw direction. If the ball is not visible or within range and the robot is localized, it aligns the upper body with the line between the goals to improve the visibility of the landmarks.

*8) Goalkeeping:* The goalkeeper's objective apparently is to keep the ball out of the own goal. It positions itself in the own goal on the line from own goal to ball, or, if the ball is not visible, on the line between the goals. Balls close to the robot let it react immediately and trigger a corresponding goalie motion to capture the ball. To achieve fast reaction on an approaching ball, the visual perception supplies an estimation of the ball velocity from two successive frames. The goalkeeper reacts on balls that have velocity larger than a fixed threshold. The type of the goalkeeper motion, i.e. bending down or diving to a side, is determined by the intersection point of the moving ball direction and the goal line.

## VIII. Tactics and Team Behaviors

The soccer behaviors so far make use of egocentric information obtained from visual perception. To implement team play and to act in an allocentric perspective, we introduced a next higher level in our behavior framework that abstracts from the reactive, egocentric soccer behaviors. On the basis of self-localization, tactical behaviors can configure the lower level soccer behaviors by setting ball-target and target pose in

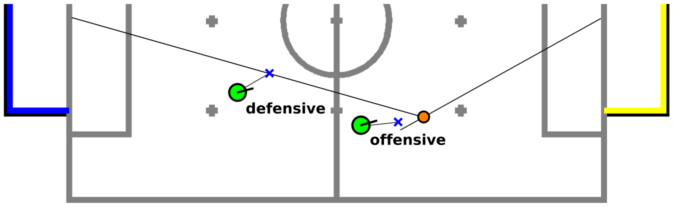



Fig. 6. An illustration for the positioning of offensive and defensive field players (green circles with orientation indicator). The offensive field player positions on the line from ball to target goal (yellow) with sagittal and lateral offsets behind the ball (orange circle) on the blue cross, such that the ball lies in front of the kicking leg. The defensive field player takes a defensive, supporting pose by positioning on the direct line from ball to own goal in a certain distance to the ball.

allocentric coordinates at a rate of 20.83Hz. To implement target-pose positioning, an additional behavior is necessary on the lower level that inhibits all other behaviors on its level. Additionally, the tactical behaviors can control, when the soccer behaviors are allowed to dribble or kick. For instance, dribbling can be preferred over kicking. In this way, special game tactics can be implemented. Also, the technical challenges of the RoboCup Humanoid League, i.e. obstacle avoidance, slalom dribbling around poles, and passing between two field players, could easily be implemented with this abstract interface.

Another main concept on this level is the role of the player within the soccer team. A player can act either as offensive field player, as defensive field player, or as goalkeeper. If only one field player is on the field, it plays offensive. When the team consists of more than one field player, the field players negotiate roles by claiming ball control. While no player is in control of the ball, all players attack. If one of the players takes control, the other player switches to the defensive role. A player may claim ball control, if it is not already claimed, relative ball position and angle are within a certain threshold, and the player is positioned better behind the ball than the other field player. As soon as the relative ball position and angle exceed a larger threshold or the player detects a fall or performed a kick, it abandons ball control. To assess the positioning quality of the players, a positioning utility is calculated by each player and communicated via WLAN to the other. The positioning utility depends on the deviation from the relative target position behind the ball that is given in the current game situation. Another application of the role concept is goal clearance by the goalkeeper: the goalkeeper switches its role to field player, if the ball gets closer than a certain distance. In this case it starts negotiating roles with other field players like a standard field player.

We implemented the following behaviors on this level, excluding special behaviors for the technical challenges:

*1) Role Assignment:* The behavior implements default role negotiation and role switching as described above.

*2) Positioning:* Positioning on a target pose is used for kick-off and for technical challenges.

*3) Ball Handling:* This behavior contains default ball handling skills, i.e. dribbling the ball close to the goal instead of kicking, and dribbling the ball from the opponent's field corner to the center of the opponent's half.

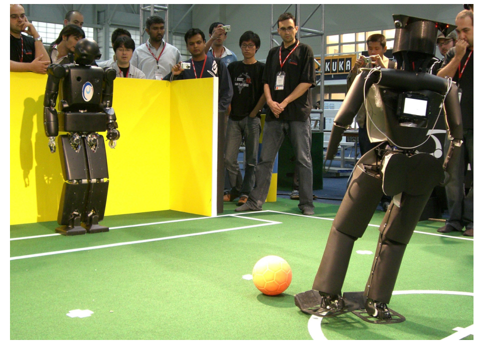


Fig. 7. Final game of the TeenSize penalty kick tournament: NimbRo vs. Pal Technology. NimbRo won the exciting match 5:4.

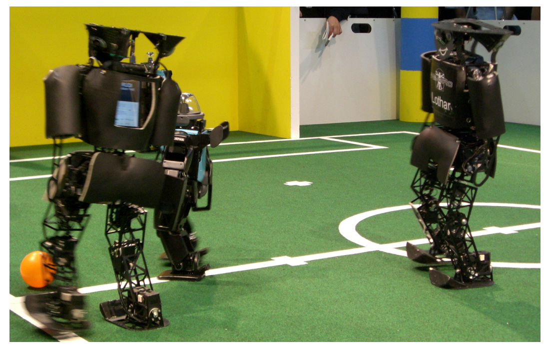


Fig. 8. Final game of the KidSize soccer tournament: NimbRo vs. Team Osaka. NimbRo won the exciting match 8:6.

*4) Dribble around Obstacles:* If the ball lies in front of a close obstacle, the player dribbles the ball around the obstacle onto the side where the angle between goal and obstacle is largest.

*5) Dribble into Freespace:* If an obstacle is located between the player and the ball-target in the opponent's half, the player dribbles the ball onto the field side that is not occupied by the obstacle.

## IX. Results

Our robots performed very well at the RoboCup 2007 competitions. In the TeenSize class, a penalty kick round robin was played between all seven teams. Robotina smoothly approached the ball and kicked it hard in one of the goal corners. Our goal keeper Bodo reacted quickly and dived to keep the goal clear. Consequently, our robots won all six games of the round robin. They scored 27 of 30 possible goals, which corresponds to a 90% success rate, and received only 10 goals. In the semi-final, they met again the titleholder Team Osaka. The game ended 4:3 for NimbRo. In the final, our robots met Pal Technology [13] from Spain (see Fig. 7). The Pal robot kicked the ball very hard, but could not dive to keep the goal clear. It adapted a strategy of walking into one of the corners to block it. Robotina was observing the goal keeper position and decided to kick the ball into the other (free) corner. The exciting game ended 5:4 for NimbRo.

Our TeenSize robot Bodo also excelled in the technical challenges. Bodo was very quick to walk across the field in the foot race. It needed only 10.27s for a distance of more than 4m. He also walked quickly though six black obstacle poles (16.46s for more than 3m), without touching any pole. Bodo dribbled the ball in slalom around three colored poles, completing two-thirds of the dribbling challenge. Team Osaka had the same degree of completion, but was faster in all three parts of the technical challenge.

In the KidSize soccer tournament, the 22 teams were split into four groups. Our robots won all games of their group. In the quarter final, they met Darmstadt Dribblers [14] from Germany. The exciting game ended 8:6 for NimbRo after extra time. Our robots met the FUmanoids [15] from Berlin, Germany in the semi-final. NimbRo won the game 11:0.

The same two teams that met in the 2005 and 2006 finals, met in the 2007 final soccer game again: NimbRo vs. Team Osaka [16] (see Fig. 8). While Team Osaka used a dedicated goalie and only one field player, we decided to use two field players and no goalie. The Osaka robots were very quick to walk behind the ball, positioned themselves for kicking, and kicked it hard across the field. The NimbRo robots excelled in one-on-one fights and in team play. When they perceived that the opponent was positioning itself for a kick, they walked against the ball to keep it moving. Thus, the Osaka robots had to approach the ball again. The second field player stayed on a defensive position, but took over, if the primary attacker lost the ball. This team play was a big advantage for our team. The exciting game was open until the end. The final score was 8:6 for NimbRo. Videos showing the performance of our robots at RoboCup 2007 can be found at http://www.NimbRo.net.

## X. Conclusion

This paper described the design of the behavior control software for our NimbRo 2007 robots, which won all humanoid soccer competitions of RoboCup 2007. We implemented the control software in a framework that supports a hierarchy of reactive behaviors. A kinematic interface for body parts made it possible to abstract from individual joints when implementing basic skills like omnidirectional walking. These basic skills made it possible to abstract from body parts when implementing more complex soccer behaviors. At this player level, our humanoid robots are very similar to wheeled or four-legged soccer robots. Finally, at the tactical level, the players of our team are coordinated through role negotiation and the soccer behaviors are configured according to the tactical situation.

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 already the largest RoboCupSoccer league. The 2007 competition has shown that most robots master the basic skills of walking, kicking, and getting up, but only few teams are able to play decent soccer games.

We expect to see the rapid progress continue as the number of players is increased to 3 vs. 3 and the field size will be enlarged in 2008. This will extend the possibilities for team play. The increased complexity of soccer games with more players will make structured behavior engineering a key factor for success.

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. In the next years, the speed of walking must be increased significantly. The visual perception of the soccer world must become more robust against changes in lighting and other interferences. We continue to work on these issues.

## Acknowledgment

This work is funded by the DFG under grant BE 2556/2-2,4.

## References

[1] H. Kitano and M. Asada, "The RoboCup Humanoid Challenge as the millennium challenge for advanced robotics," *Advanced Robotics*, vol. 13, no. 8, pp. 723–737, 2000.

[2] S. Behnke and R. Rojas, "A hierarchy of reactive behaviors handles complexity," in *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Springer, 2001, pp. 125–136.

[3] S. Lenser, J. Bruce, and M. Veloso, "A modular hierarchical behavior-based architecture," in *RoboCup 2001: Robot Soccer World Cup V*, ser. LNCS, vol. 2377. Springer, 2002, pp. 423–428.

[4] H. Jaeger and T. Christaller, "Dual dynamics: Designing behavior systems for autonomous robots," *Artificial Life and Robotics*, vol. 2, no. 3, pp. 108–112, 1998.

[5] J. Hertzberg and F. Schönherr, "Concurrency in the DD&P robot control architecture," in *Proceedings of Symposium Engineering of Natural and Artificial Intelligent Systems (ENAIS)*, 2001.

[6] A. Bredenfeld and H.-U. Kobialka, "Team cooperation using dual dynamics," in *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Springer, 2001, pp. 111–124.

[7] H. Utz, G. Kraetzschmar, G. Mayer, and G. Palm, "Hierarchical behavior organization," in *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Edmonton, Canada, 2005, pp. 2598–2605.

[8] M. Lötzsch, J. Bach, H.-D. Burkhard, and M. Jüngel, "Designing agent behavior with the extensible agent behavior specification language XABSL," in *RoboCup 2003: Robot Soccer World Cup VII*, ser. LNCS, vol. 3020. Springer, 2004, pp. 114–124.

[9] T. Thomas Röfer, "A behavior architecture for autonomous mobile robots based on potential fields," in *RoboCup 2004: Robot World Cup VIII*, ser. LNCS, vol. 3276. Springer, 2005, pp. 122–133.

[10] S. Behnke, "Online trajectory generation for omnidirectional biped walking," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'06), Orlando, Florida*, 2006, pp. 1597–1603.

[11] J. Stückler and S. Behnke, "Soccer behaviors for humanoid robots," in *Proc. of the Workshop on Humanoid Soccer Robots of the IEEE-RAS Int. Conf. on Humanoid Robots, Genoa, Italy*, 2006, pp. 62–70.

[12] J. Stückler, J. Schwenk, and S. Behnke, "Getting back on two feet: Reliable standing-up routines for a humanoid robot," in *Proc. of Intelligent Autonomous Systems (IAS-9), Tokyo, Japan*, 2006, pp. 676–685.

[13] D. Faconti, "Technical description of REEM-A," in *RoboCup Humanoid League Team Descriptions, Atlanta, GA*, 07/2007.

[14] M. Friedmann, J. Kiener, S. Petters, D. Thomas, and O. von Stryk, "Darmstadt dribblers," in *RoboCup Humanoid League Team Descriptions, Atlanta, GA*, 07/2007.

[15] H. R. Moballegh, R. Guilbourd, G. Hohl, A. Reuschenbach, J. Yu, and R. Rojas, "Fumanoid team description 2007," in *RoboCup Humanoid League Team Descriptions, Atlanta, GA*, 07/2007.

[16] R. Matsumura, N. Shibatani, T. Imagawa, T. Maeda, T. Miyashita, T. Takahashi, Y. Akazawa, N. Yamato, and H. Ishiguro, "TeamOSAKA (Kid size)," in *RoboCup Humanoid League Team Descriptions, Atlanta, GA*, 07/2007.