B-Human

Team Description for RoboCup 2008

Thomas Röfer[1], Tim Laue[2], Armin Burchardt[2], Erik Damrose[2], Martin Fritsche[2], Judith Müller[2], and Andrik Rieskamp[2]

[1] Deutsches Forschungsinstitut für Künstliche Intelligenz GmbH,
Sichere Kognitive Systeme, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
E-Mail: `thomas.roefer@dfki.de`

[2] Fachbereich 3 - Mathematik und Informatik, Universität Bremen,
Postfach 330 440, 28334 Bremen, Germany
E-Mail:
`{timlaue,armin,edamrose,fritsche,judy,rieskamp}@informatik.uni-bremen.de`

1 Introduction

Team B-Human consists of computer science researchers and students mainly from the University of Bremen. The team was founded in 2006 with a small amount of members. In 2007, it was strengthened by further computer science students who support the team on the basis of their advanced study project. Due to this distribution of expertise, our team is primarily focused on the software components rather than on the robot construction. At the very beginning, the approach of the team was to transfer as much code and experiences from the Four-Legged League to the Humanoid League as possible [1, 2]. However, by now the team has made lots of own experiences, has rewritten most of the code, and stands on its own feet.

Our first competition was the RoboCup German Open 2007 in Hanover, where we reached the 4th place. At the RoboCup 2007 in Atlanta, we were undefeated in the preliminaries with a goal ratio of 7:1, and reached the quarterfinals. Hence we were able to demonstrate that our robots can actually play soccer. Nevertheless we learned a lot in these first competitions and improved significant part of our hard- and software. For the year 2008, we plan to participate in the RoboCup German Open as well as in the World Cup.

2 Robot Hardware

Our current robot model was designed and constructed in 2006 and 2007. The platform is based on the *Robotis Bioloid* commercial standard robot construction kit, which has already been used in RoboCup competitions by some other teams. It carries a low-level controller board for communication with the actuators and sensors. The platform has been upgraded with a voltage regulator for more reliable power supply and an additional sensor board assembled with a 3-axis gyroscope, 3-axis acceleration sensor, and a magnet compass.

Fig. 1. Jen – the robot scored six goals at RoboCup 2007

Each of our robots is equipped with a standard PDA from Fujitsu Siemens running Microsoft Windows Mobile which is responsible for the high-level onboard computations. The PDAs have an integrated 1.3 mega pixel camera which provides us with 15 YUV422 images per second with a resolution of 640x480 pixels, of which we use 320x240 YUV pixels.

For more details on the robot hardware see the specifications page.

3 Software Architecture

B-Human is using the software framework of the GermanTeam 2007 [3] as a basis for its control software. The framework is the successor of one of the most popular robot control architectures for RoboCup teams [4]. Large parts of the included components, tools and the behavior description language XABSL [5] have already been used in several other RoboCup leagues such as the Four-Legged League, the Middle-Size League and the Small-Size League. The latest version of the architecture provides powerful mechanisms for structuring and debugging. In general, it structures the code into *modules*, *representations*, and *processes*. A *module* solves a specific task and is encapsulated by a well-defined interface consisting of *representations*. Each module requires some representations as input and provides other representations as output. The developer can dynamically select the set of modules that are active. The modules are automatically ordered so that each module is only executed after all its required representations have been updated, and the selection of modules is checked for consistency. *Processes* run concurrently and group modules together. They define whether modules are executed sequentially (if they are part of the same process) or run concurrently (if they are part of separate processes).

A well-established approach (cf. Fig. 2) of grouping all tasks required to play soccer into one process running at video frame rate (*Cognition*) executing all modules of cognition and behavior control, and another one running at the fre-

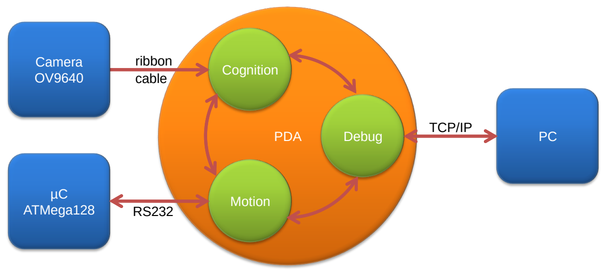

Fig. 2. The processes running on the PDA

quency required for sending the motion commands (*Motion*). Our architecture contains another process (*Debug*) which distributes and collects debugging information and communicates them with an off-board PC. This process is only active during software development and is inactive during actual RoboCup games.

4 Modules

The modules that enable our robots to play soccer can be structured into five different categories:

Infrastructure. The modules of this category communicate with the actual hardware of the system and provide the readings as representations.

Perception. The current states of the joints are analyzed to determine the position of the camera. The camera image is searched for objects that are known to exist on the field, i. e. goals, flags, field lines, and the ball. In addition, the acceleration sensors are employed to determine whether the robot fell down.

World Modeling. Percepts immediately result from the current sensor readings. However, most objects are not continuously visible, and noise in the sensor readings may even result in misrecognition of an object. Therefore, the positions of the dynamic objects on the field have to be modeled, i. e. the location of the robot itself, the position of the ball, and the free space on the field.

Behavior Control. From the information provided by perception, world modeling, and the role of the robot, the behavior of the robot is generated. It is described as a hierarchical state machine in XABSL [5], and it decides which motions have to be executed by the head and by the whole body.

Motion Control. The motions requested by the behavior control such as walking, standing up, kicking, and cheering are executed and coordinated by modules of this category. There is also a feedback about the motion currently executed and its effect on the position of the robot (odometry).

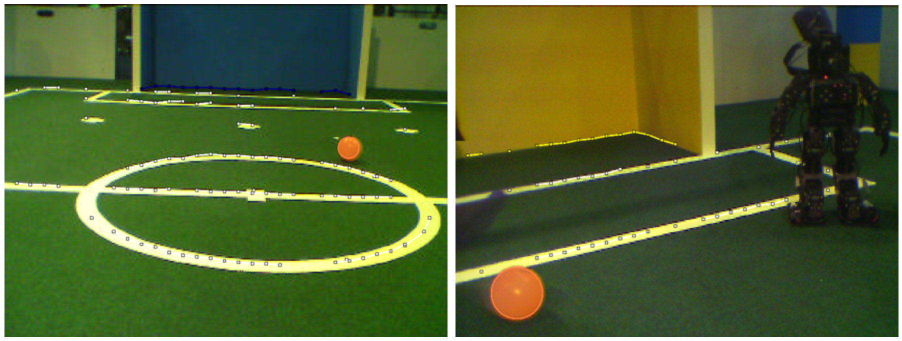

Fig. 3. Two images from the robot's camera. Some drawings in the images depict perceived elements of the environment: orange circles for balls, white dots for field line elements, and yellow and blue dots for the lower edges of goals.

5 Vision and State Estimation

Our Bioloid robots use a directed vision system based of the PDA's original camera (lead outside the case) which is mounted on a pan-tilt unit. This camera provides 15 images per second with a resolution of 320 by 240 pixels; its opening angles are 45.1°by 34.8°. This is a quite limited field of view, as apparent in Fig. 2. To keep track of the own pose as well as of the ball's position, the camera needs to be moved constantly by the pan-tilt unit.

Out vision software is able to extract some basic features that provide the base for the localization algorithms. These features (i.e. the ball and significant points on the field) are depicted in Fig. 3. Their recognition is based on the detection of significant color changes in the picture. Since the camera's perspective is known, it is possible to differentiate between features which actually belong to the field and those that don't (e.g. spectators). Nevertheless, the computation of the feature positions relative to the robot is subject to heavy noise, since the camera is on top of a walking (and shaking) robot. This needs to be compensated afterwards by the localization algorithms.

For self-localization, B-Human uses a particle filter based on the Monte Carlo method [6] as it is a proven approach to provide accurate results in such an environment [7]. Additionally, it is able to deal with the kidnapped robot problem, which often occurs in RoboCup scenarios. For a faster reestablishment of a reasonable position estimate after a kidnapping, the *Augmented MCL* approach by [8] has been implemented.

The estimation of the ball's position and velocity has been realized in a similar way. Figure 4 shows an example of these two probability distributions. A comprehensive description of our state estimation implementation is given in [9].

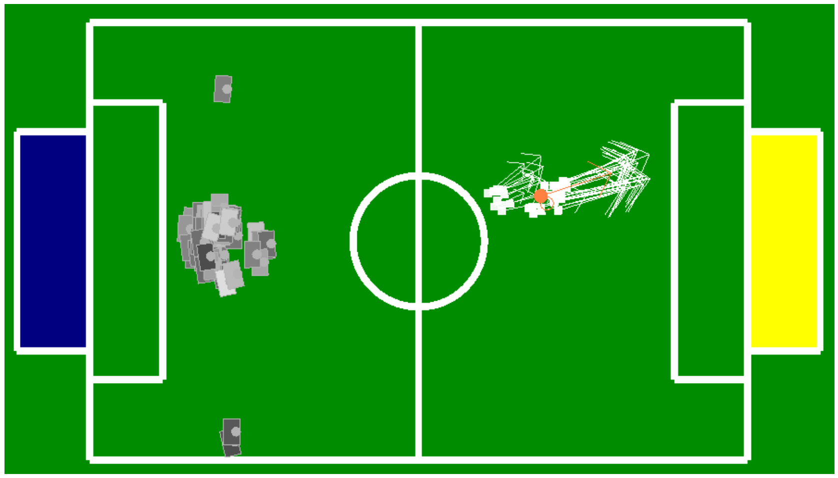

Fig. 4. Illustration of particle-based representation of the robot pose and the ball position and velocity. Each gray box denotes one possible pose of the robot; light boxes are more likely than dark boxes. The black box shows the resulting robot pose from the filter. The dots with arrows describe the ball samples (describing position and velocity of a ball). The orange one is the resulting ball estimate.

6 Robot Motion

We distinguish between two kinds of motion: On the one hand we have our walking engine which allows us to perform omni-directional movement as it is desirable to be able to move in any combination of forward, sideward, and rotational direction in soccer scenarios. To support the omni-directional movement our robot carries six servos per leg: Three in the hip (roll, pitch, yaw), one in the knee and two to control pitch and roll of the foot. To increase the walking stability we incorporate the data of the acceleration sensors to control the body tilt and sideward swinging.

On the other hand we have so-called special actions in order to be able to perform a sequence of sets of pre-defined joint positions. Examples for special action usage is kicking and standing up. During the execution of a special action the joint angles are either interpolated to allow fluid movements or they are simply set, ignoring the previous values of the servos to reach maximal strength, i. e. a kick.

7 Current Work and Future Plans

We are currently building and testing a new prototype, which has some improvements over the old model. The approach is to save weight in order to allow faster and more stable movements. Since the PDA computing units of the robots have reached their limit in terms of computing power, we are planning to replace them with a PC104 board, which is currently being tested. Other activities include gait optimization with a Particle Swarm Optimization algorithm (which has already been carried out successfully on a different robot [10]), testing the use of a goal keeper, improving the vision system and modeling behaviors for the technical challenges.

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft in context of the Schwerpunktprogramm 1125 (*Kooperierende Teams mobiler Roboter in dynamischen Umgebungen*).

References

1. Laue, T., Röfer, T.: Getting Upright: Migrating Concepts and Software from Four-Legged to Humanoid Soccer Robots. In: Proceedings of the Workshop on Humanoid Soccer Robots in conjunction with the 2006 IEEE International Conference on Humanoid Robots. (2006)
2. Röfer, T., Fritsche, M., Hebbel, M., Kindler, T., Laue, T., Niehaus, C., Nistico, W., Schober, P.: BreDoBrothers - Team Description for RoboCup 2006. In: RoboCup 2006: Robot Soccer World Cup X. Lecture Notes in Artificial Intelligence, Springer (2006)
3. Röfer, T., Brose, J., Göhring, D., Jüngel, M., Laue, T., Risler, M.: GermanTeam 2007. In Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F., eds.: RoboCup 2007: Robot Soccer World Cup XI Preproceedings, RoboCup Federation (2007)
4. Röfer, T.: An architecture for a national robocup team. In: RoboCup 2002 Robot Soccer World Cup VI, Gal A. Kaminka, Pedro U. Lima, Raul Rojas (Eds.). Number 2752 in Lecture Notes in Artificial Intelligence, Springer (2003) 417–425
5. Lötzsch, M., Bach, J., Burkhard, H.D., Jüngel, M.: Designing agent behavior with the extensible agent behavior specification language XABSL. In Polani, D., Browning, B., Bonarini, A., Yoshida, K., eds.: RoboCup 2003: Robot Soccer World Cup VII. Number 3020 in Lecture Notes in Artificial Intelligence, Padova, Italy, Springer (2004)
6. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In: Proc. of the National Conference on Artificial Intelligence. (1999)
7. Röfer, T., Laue, T., Thomas, D.: Particle-filter-based self-localization using landmarks and directed lines. In: RoboCup 2005: Robot Soccer World Cup IX. Lecture Notes in Artificial Intelligence, (Springer)
8. Gutmann, J.S., Fox, D.: An experimental comparison of localization methods continued. Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2002)
9. Laue, T., Röfer, T.: Particle Filter-based State Estimation in a Competitive and Uncertain Environment. In: Proceedings of the 6th International Workshop on Embedded Systems, VAMK, University of Applied Sciences; Vaasa, Finland (2007)
10. Niehaus, C., Laue, T.R.T.: Gait Optimization on a Humanoid Robot using Particle Swarm Optimization. In Zhou, C., Pagello, E., Menegatti, E., Behnke, S., eds.: Proceedings of the Second Workshop on Humanoid Soccer Robots in conjunction with the 2007 IEEE-RAS International Conference on Humanoid Robots. (2007)