Humboldt Heroes

Matthias Werner, Helmut Myritz, Uwe Düffert,
Martin Lötzsch, Hans-Dieter Burkhard

Institute of Computer Science, Humboldt University of Berlin

1 Introduction

Our last year's approach to the Sony legged league suffered from a number of drawbacks:

- The motion system was ways to slow and imprecise.
- The different tasks were not well coordinated. The robot got quite a number of deadlock situations and was not real-time at all.
- The controlling based on a pure reactive approach, that did not allow any real planing.

With this year's code, we tried to overcome some of these disadvantages.

We partly succeed. We get second in group and went to the final round. However, we never had any chance in the quarter final against the cup-dominating UNSW team.

2 Team Development

Team Leader: Hans-Dieter Burkhard
Team Members:

Matthias Werner
- Assistant professor
- Technical leader, also responsible for the run-time system
- Did attend the competition

Helmut Myritz
- Graduate student
- Responsible for AI and self-localization
- Did attend the competition

Andrej Georgi
- Graduate student
- Responsible for visioning
- Did not attend the competition

Uwn Düffert
- Undergraduate student
- Responsible for locomotion
- Did attend the competition

Martin Lötzsch
- Undergraduate student
- Responsible for tool support and locomotion
- Did attend the competition

Web page http://www.informatik.hu-berlin.de/ mwerner/res/robocup/index.en.html

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 651-654, 2001.

3 Architecture

Our soccer software for the ERS 110 consist of 4 active modules, which are supported by runtime system:

- Central control
- Perception unit
- Localization system
- Motion unit

The runtime system encapsulates each module as an Aperios object and activate them in a time-triggered manner. All modules communicate among each other via a shared memory.

All players share the identical program. The information about player's role in the game, direction of attack, etc. are coded in a configuration file which is parsed during the robot's initialization.

4 Vision

Our vision system bases on the color detection provided by Sony. To generate the needed color tables, we use a semi-automatic tool, which allows to construct appropriate filter data for each color on base of a number of sample pictures.

To determine the distance and direction of other robots, we do an iterative pattern matching of the gained data against a projection of a simplified robot model. Since this procedure is quit time-consuming, the search is invoked for close robots only (large red or blue spots).

Unfortunately, our visioning system turns out to be to sensible against small changes of light. Also it had quite a few difficulties to distinguish between yellow and orange.

5 Localization

To solve the problem of localization we decided to use a case-based reasoning approach. In this approach the robot imitates the remembering process of humans.

The robot uses a given case-base consisting of 126 (14 x 9) cases, corresponding to a 14 by 9 grid that covers the pitch. Thus, robot knows for each case the correct position on the field, the angles to all landmarks, and the seen sizes of these landmarks.

With this knowledge the robot can now try to localize itself on the pitch with the granularity of the grid. During the localization our robot has to stand safely, without any movement. His head now scans the whole environment by turning to left an right.

During this scan all information about all seen landmarks (flags and goals) are stored and processed through a CRN (Case Retrieval Net). Finally, the information is used to activate different cases in different priority. If succeed, the position of the case with the highest priority describes a position which is close to the real position of the robot on the pitch.

During the interception of the ball the robot uses a relative world model, where it position is described relative to the ball. The described localization procedure is invoked when the robot successfully intercepted the ball.

6 Behavior Control (BDI)

For the behavior control we use the BDI approach that was already successful in the simulation league.

The BDI-Model (Belief-Desire-Intention) can easily be described by following steps of a decision process:

- At first the robot tries to collect as much information about his environment as possible (position by CRN, informations about ball and other players) and stores it in its world model (belief phase).
- After that, it searches for all options (e.g.: InterceptBallOption, GoalKick-Option ...) and determines, which of them is the option with the highest priority (desire phase).
- Now, the robot will realize the desired option, by executing a plan assigned to this option (for example: walk 2 steps in front and then kick) (intention phase).

Finally, the robot is ready to handle the new environment informations and the process will repeat. The decision component was invoked two times a second. If a new plan comes up during the execution of an old one, the robot tries to calculate the cost of an abort of the old plan. To stabilize the decision process, every change of a non-finished plan includes a minimum cost, even if the new plan could be easily be integrated.

7 Action/Walking

Thinking of the troubles with not completely known time behaviors, dependencies and parameterizing possibilities of the motions given in OMoNet we decided to implement our own motions.

Therefore, we developed a high level language that can describe a motion net quite similar to OMoNet and a flex/bison based compiler called MotionConfigTool that produces C++ source code from motion net descriptions in that language, which can be compiled and linked into Aibo binaries. The compiler checks a lot of dependencies, such as the existence of transitions between all motions, so that we had a very robust solution for our motions on the Aibo.

Motions can be defined in different ways. Some simple motions (like a Getup or our Headkick[1]) can fully sufficient be defined as a small number of vectors

[1] The idea of headkick is got form the ARAIBO team.

with values for destination positions for each joint of the dog and with a time, these positions have to be reached in.

For all walking-like motions (walking and turning in all directions) we had to use a more complex approach. We consider a number of three-dimensional constants or variables like shoulder height or side distance between body and feet. To make the dog move we simulate the moving of the center of gravity of the dog while keeping all feet on the ground and calculate the resulting arcs for all 12 joints.

Of course we have to lift some legs from time to time to make steps. At the moment we make a step by letting a foot move on a half circle from the old position to the simulated position the foot should have a whole step later.

We recognized that leaving all feet on the ground half the time and rising 2 diagonal legs to make a step the other half gives a quite stable walk although the center of gravity cannot be between three legs touching the ground at all times.

Acceleration is not considered in our model yet and because it is hard to simulate it and easier to minimize the occuring divergence by correcting existing parameters, we will probably never implement it.

8 Conclusion

Our team's biggest drawback was the color recognition and blob separation. Thus, next year we will not use Sony's CDT anymore but develop our own color detection system.

Also, we have the intension to improve the following features:

- **Motion system.** Acceleration of the movement, more flexible model to allow non-horizontal positions
- **AI.** Multi-level BDI to allow long-range intentions
- **Self-Localization.** Additional to the CRN, we want to use an odometric system.

Overall, there was an significant improvement in handling the robot, compared with the RoboCup99. We hope that we can continue our work and successful take part in RoboCup2001.