Middle Robots League, Team *Team-Tuebingen*, pages 200–207
http://www.ep.liu.se/ea/cis/1999/006/32/

The Attempto RoboCup Robot Team

Team-Tuebingen

Michael Plagge, Richard Günther, Jörn Ihlenburg, Dirk Jung, Andreas Zell

MP, RG, JI, DJ, AZ: University of Tübingen

Abstract. *This paper describes the hardware and software architecture of the Attempto RoboCup-99 team. The different parts of robot, sensor and computer hardware which are used are presented. Then the robot control architecture which realizes a hybrid control, consisting of a reactive behavior based component and a planner component for more complex tasks is introduced. Also the problems we currently are working on are presented, as there are a fast and reliable self localization algorithm and a robust behavior based reactive component for the hybrid control system.*

1 Introduction

For building a good team of agents that can take part in the RoboCup-99 contest, ideas and results from different fields of research, e.g. artificial intelligence, robotics, image processing, engineering, multi agent systems can or even must be used and tested [Kitano *et al.*, 1997],[Asada, 1998].
Since we are developing a team for the mid-size contest our main objective is to build a robot system, which is able to recognize the environment in a suitable way and to build a fast and reliable control system, which is capable of solving the given task of playing football. This control system must cope with the dynamics and adverse aspects of RoboCup-99 and with complex situations, e.g. teamwork which occurs in RoboCup-99.
The remainder of the paper is structured as follows: Section 2 describes the robot, sensor and computer hardware of the Attempto team robots. Section 3 gives an overview about the three different layers of our software architecture. Section 4 focuses on our concepts to address the problems of doing a reliable and fast self localization and to develop a fast and robust reactive control component.

2 Hardware

2.1 Robot platform

As the basic robot platform for the field player we are using the Pioneer2 DX from ActivMedia Inc. (Fig. 1). This robot is equipped with a differential drive system with a free running caster wheel mounted at the back of the robot. The maximum achievable translation speed is about 1,5 m/s, the maximum rotational speed is 2 π/s. The robot can carry weights up to 20 kg and can be equipped with a maximum of three 7,2 Ah batteries, which allows an operating time with all additional hardware like PC and sensors of nearly three hours without recharging.
The two driving motors are equipped with 500 tick position encoders. With these encoders the speed and the position of the robot can be obtained. The robot is controlled by a Siemens C166 microcontroller. This device is responsible for controlling the actuators of the robot and for the calculation of the position and orientation from the motor encoder data.
Via a serial device the controller can communicate with a remote computer. This device can operate at a maximum speed of 38400 bauds. The robot sends 20 times a second a status data packet to the remote computer. It also accepts commands from the remote computer with the same rate. Therefore the minimal achievable response time for a closed loop controller is about 50ms.
As the basic platform for the goalkeeper we are using a Pioneer AT. Each of the four wheels of this robot is driven by its own motor. The wheels on each side are coupled with a belt. The battery with a capacity of 12Ah allows an operating time with our additional hardware of 1.5 hours. A custom designed board with a MC68332 CPU replaces the standard MC68HC11 board and gives faster response time, higher precision of odometry and more flexible sonar firing patterns. Despite serious problems in the preliminary rounds the goalkeeper was influential for our 1998 success at Paris reaching the final.

2.2 Sensors and actuators

As we are convinced that better sensors will result in a better situation assessment and, ultimately, in better playing capabilities, we try to employ a diversity of sensors on the robot. While the final design is not finished at the time of this writing, we are considering the use of the following sensors: Sonars, 2d laser scanner, IR sensors, colour camera, 360^0 camera, digital compass.
Sonars: The Pioneer2 DX is equipped with eight, the Pioneer AT with seven Polaroid 6500 Ultrasonic transducers, which are mounted in front and at the front side of the robot.
IR Sensors: The second type of distance measurement sensor are infrared sensors, which works with a triangulation method for measuring the distance to an object. The sensing range is from 10 to 80 cm. The measured distance to an object is nearly independent from the reflectivity of the surface of this object. The sensors will be mounted at the back and the rear sides of the robot, to allow the robot to move backwards.
Laser scanner: If it is possible to mount a laser scanner in a suitable way on the robot without interfering with either sonars, cameras or kicker we

will perhaps also use these devices with our robots. The employed laser scanner would be a LMS200 from SICK AG. It has a 180^0 field of view and a angular resolution of $0{,}25^0$. It can measure distances up to 15 m with an accuracy of 10 mm. With a resolution of 1^0 and a total field of view of 180^0 and 500 kbps data transfer rate over a RS422 serial device the achievable scan rate is nearly 60 Hz. This sensor, which is a successor to the device which secured Freiburg's [Gutmann *et al.*, 1998] advantage last year, is currently the fastest and most precise distance measurement device. Its main drawbacks are its size (137*156*185 mm), weight (4,5 kg) and power comsumption (max. 17,5 W).

Color camera: For the task of object detection and classification we are using two vision systems. Both systems use a Siemens SICOLOR C810 CCD-DSP color camera, with a 1/3 inch CCD-chip and a resolution of 752x582 pixel. The output format is a regular CCIR-PAL signal with 625 rows and 50 half frames per second. One of the cameras is mounted at the front of the robot. This camera is equipped with a 2,8f wide angle lens. It is mainly for the detection of the ball and the objects, which lie in front of the moving robot. This camera is also responsible for distinguishing team mates from opponent robots.

360^0 camera: The second camera is mounted in an omnidirectional vision system, which is mounted at the top of the robot (Fig. 2). A 4,2f lens is mounted at this camera, to achieve a large visual field. The design of this camera has been made by Matthias Franz from the MPI for Biological Cybernetics from an earlier MPI design used for biologically inspired vision experiments. In contrast to most other omnidirectional vision systems this design has a paraboloid mirror instead of a conical mirror. This should give a better mapping of objects below the horizon.

Digital compass: This device is capable to determine the absolute orientation of the robot, where the error in measurement does not depend on the distance traveled or on other influences the odometry suffers from. It sends heading data with 5Hz, a resolution of 1^0 and an accuracy of 2^0.

Kicker: We plan to adapt the pneumatic kicking device used at the RoboCup-98 contest in Paris to the new robots. This kicker consists of a pneumatic cylinder, an electric valve, and a tank for compressed air. We also developed a second kicking device based on a spring mechanism wound up by a BMW car windshield wiper motor. This spring loaded kicker even shot harder than the pneumatic kicker and was successfully demonstrated at the Vision RoboCup-98 at Stuttgart, but it could not easily be fitted into the P2 chassis.

2.3 Onboard Computer

The onboard computer is the same as the one used last year in Paris, with the exception of an improved power system (the old suffered several failures in Paris and Stuttgart). It is a custom design based on standard PC parts with custom enclosure and is mounted at the rear top of the robot. Each PC has a 166 MHz Intel Pentium CPU, 64 MB RAM and a 1,2 GB Hard Disk Drive. Additionally each computer is equipped with two PCI framegrabbers with a Booktree BT484 chip. These devices deliver images in YUV-format at 25 fps (PAL) and a maximum resolution of 768x576 pixels. For the connection to the laser scanner a high speed RS422 serial card was modified to achieve a data rate of 500 kbps, the highest data rate supported by the

Figure 1: The Pioneer2 robot

Figure 2: The omnidirectional vision system

laser scanner. For the communication between the robots and to an external file server wireless PCMCIA Ethernet cards in a PCMCIA to ISA adaptor from ARtem Datentechnik, Ulm, with a data transfer rate of 2 Mb/s are used. For this device we also developed a Linux device driver, which has now found its way back to the sponsor.

3 Software architecture

The software architecture of the Attempto team can be divided in three different layers (Fig. 3): low level data processing, intermediate level layer, high level robot control. We now describe each layer in detail.

3.1 Low level data processing

In the bottom layer different server programs organize the communication with the sensor and robot hardware, and do the first steps of data processing.
The robot server receives status data from the robot, which contains position, wheel velocity, sonar data, and battery status and sends movement commands to the robot, which are received from the Arbiter. This server also receives the orientation data from the compass. The aim in developing

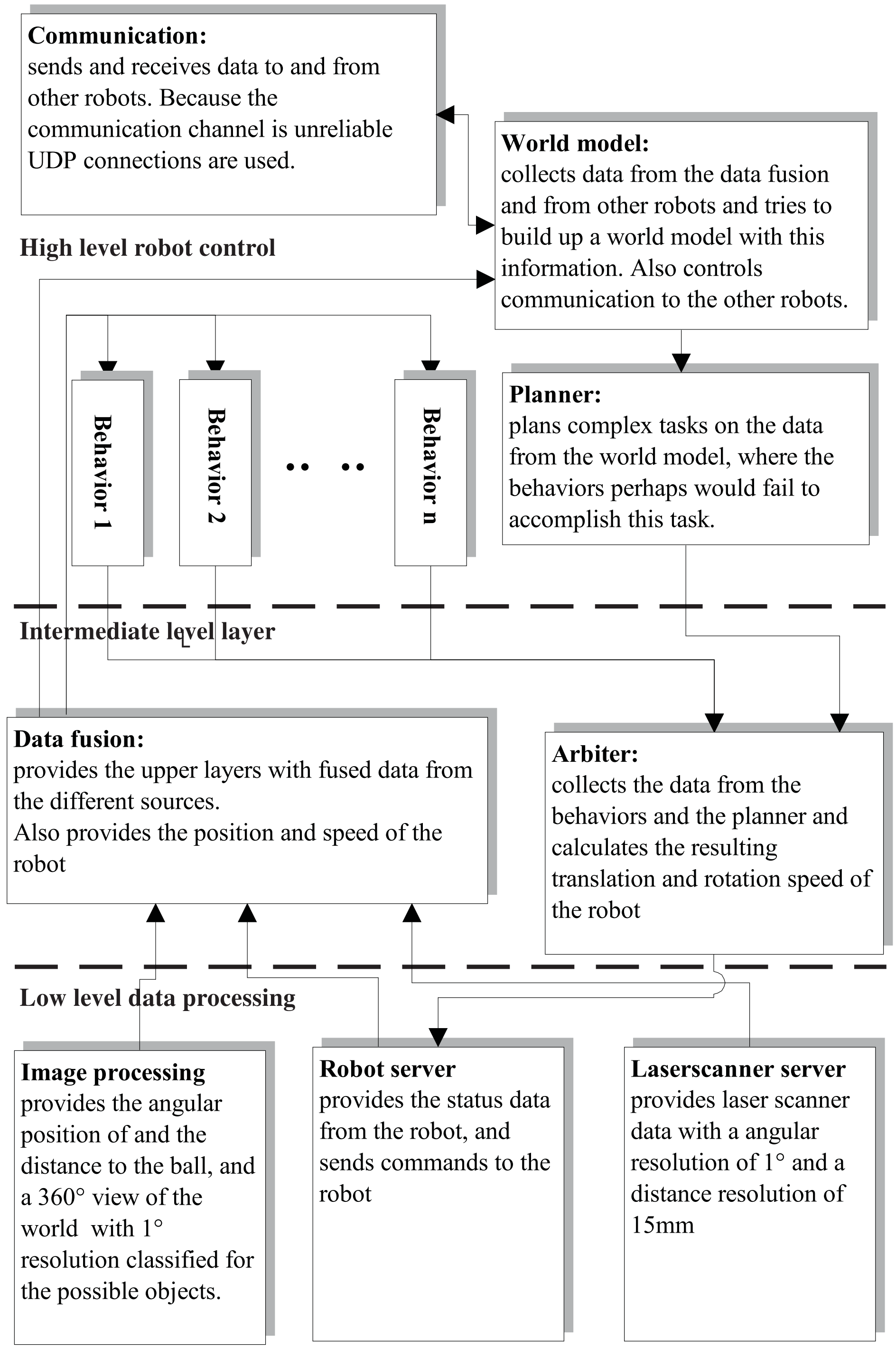

Figure 3: layered software architecture of the Attempto RoboCup-99 robots

the robot server was to send commands as fast as possible to the robot, under the constraint that the robot is only capable to execute 20 commands per second, to achieve a minimum duration for one control loop cycle.
The laser scanner server configures the laser scanner device at startup time with a field of view of 180^0, an angular resolution of 1^0 and a distance resolution of 10mm. The laser scanner then starts to send whole 180^0 scans with a rate of 60 Hz.
The image processing grabs images from the front camera and the omnidirectional camera with a resolution of 384*288 pixels in YUV format. With this resolution it is possible to detect the ball with the front camera over a distance of 8 m and estimate the ball size and therefore the ball distance with an accuracy of 5 percent. The error in the angular position estimate is less than 1 degree.
To save processing time, the image processing does not search the whole image for the ball, but uses a history of ball positions in old images to predict the position of the ball in the next image. Only if the ball is not at the predicted position, the whole image is searched, starting the search at the predicted position.
Aside from the ball position the image processing provides an array data structure with 360 elements. Each of this elements represents a field of view of 1^0 and contains information about detected objects (ball, robots, goal, wall) and the determined distance and distance errors of these objects. In the field of view of the front camera the data structure additionally contains information about the type of the detected robots (own or opponent).
Our high speed image processing needs only 3 ms per frame in the worst case (ball not at the predicted position). The average processing time for one frame is less than 1 ms. Therefore the image processing is capable of handling the 2 * 25 fps which the framegrabbers write to main memory in real time.

3.2 Intermediate layer

The intermediate layer consists of two different modules, the data fusion module, which fuses the data from the different low-level data processing servers and the arbiter, which receives steering and control commands from the behaviors and the planner and calculates a resulting movement command for the robot (section 4.2). The data fusion reduces the amount of information by extracting relevant object data from the raw sensor data. Objects fall in two different classes: dynamic objects like the ball and the other robots and static objects like the walls and the goals. The extracted information about an object includes opening angle in the field of view, distance and type of the object. The estimation error in the distance measurement is provided [Mojaev and Zell, 1998]. Therefore for the upper layer it is not necessary to know from which sensor source a specific distance measurement comes, because the properties of the sensor device are modeled via the measurement error. The data fusion also fuses the status data of the robot. For this reason, it receives the data from the odometry and compass and tries to adopt the position of the robot with the information from the self localization algorithm described in section 4.1.

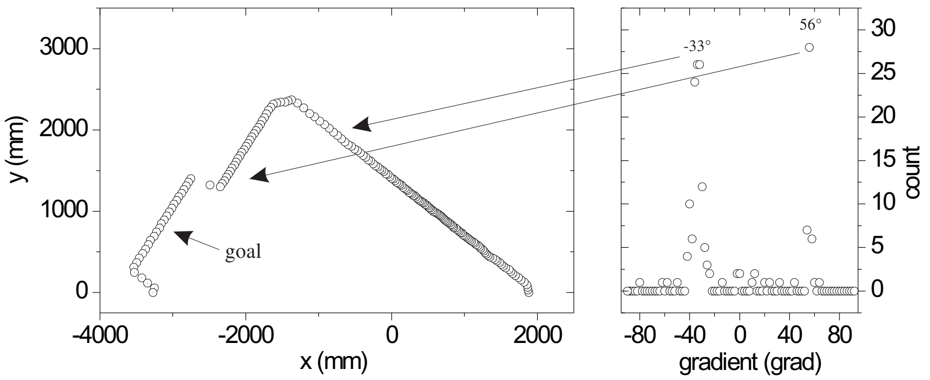

Figure 4: a) laser scan of a part of our RoboCup field. b) the corresponding histogram for the directions of the difference vectors between two scan points. The two arrows point to the lines which correspond to the maximums in the histogram

3.3 Top layer

The top layer realizes the hybrid robot control architecture [Arkin, 1998]. It consists of a reactive component where a set of independent behaviors like obstacle avoidance, ball search or ball following try to fulfill their tasks. The behaviors can react quite fast on changes in the environment because they work directly on the preprocessed sensor data. This system is easy to expand because it is possible to start and stop behaviors at runtime.
The planner component is responsible for resolving more complex situations. This component is capable of suppressing or enhancing the output from specific behaviors and can also work as a special behavior with the same output to the arbiter like the other behaviors. The planner works on the data from the world model. This module fuses the data from the internal sensors and the data coming from other teammates via the wireless Ethernet connection. It tries to keep track and identify all the objects in the environment, and tries to predict the trajectories of recently undetected objects. This component is also responsible for sending data of all objects detected by the internal sensors to all the other robots. At the time of this writing not all details of the planner have been determined.
The communication channel over the wireless Ethernet connection to the other robots is unreliable. Therefore we are using a UDP based protocol to prevent a communication action from locking while waiting for another robot to acknowledge.

4 Research Topics

In this section we give a brief overview of some of the problems we are currently working on.

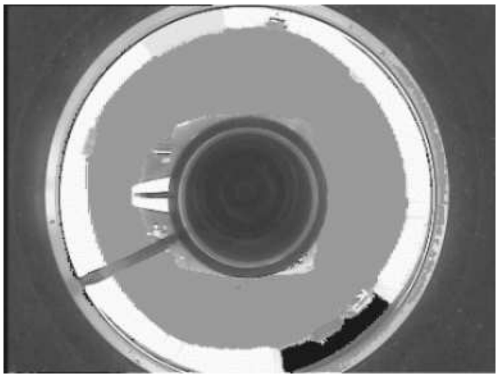

Figure 5: Image of the omnidirectional vision system.

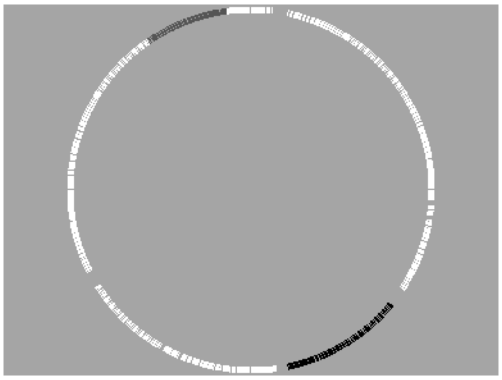

Figure 6: Classified objects from the omnidirectional view in figure 5 (white: wall, grey: the two goals)

4.1 Fast self localization with fused sensor data

Perhaps the most important information a robot needs to know to operate successfully in the RoboCup-99 is his own position within the field [Gutmann *et al.*, 1998]. Therefore a fast self localization algorithm was developed, which makes use of the fused sensor data. In a first step this algorithm calculates the vectors and the directions between successive points in a laser scan. Then a histogram is calculated for these directions [Weiss and v. Puttkammer, 1995]. In a polygonal environment like the RoboCup-99 field this histogram shows usually one or more maxima which correspond to the main directions in the environment (Fig. 4a,b). Now the laser scan is segmented into lines, by projecting each of the normalized difference vectors onto the unit vector in the main direction or directions. If the result exceeds a certain threshold the two points cannot lie on the same line. After the segmentation there is a set of lines with directions according to the maxima in the histogram. The problem now is to decide, whether some of these lines correspond to a wall or a goal in the RoboCup-99 environment, because if this is the case, the distance to this wall can be used to adopt the robot position. Especially if it is possible to find two lines on different walls, the global position within the field can be calculated. For the case that there is additional visual information from the omnidirectional vision system, its quite easy to solve the problem. It must only be checked that there is no

dynamic object between the robot and the wall or goal. That means that by fusing the data from different sources a very fast and effective self localization algorithm can be implemented.
At the moment we are also working on a self localization algorithm which purely relies on the data from the omnidirectional vision system. This algorithm works with a set of snapshots and tries to match the current view (Fig. 5,6) of the environment against these snapshots to determine the actual position with respect to the positions, where the snapshots were taken [Franz *et al.*, 1998].

4.2 Hybrid robot control architecture

To play in the RoboCup-99 environment means to fulfill a quite complex task in a dynamic, adverse environment. Therefore the robot is equipped with hybrid control architecture, existing of a reactive component and a planning component. A set of behaviors realize the fast, reactive part which is capable of dealing with the aspect of the dynamic and adverse environment. The planner should control more complicated tasks, where a purely reactive control could fail, e.g. team cooperation. The problem in question is to find a suitable way to merge the different outcomes of the behaviors and the planner. For this problem some solutions were proposed, e.g. the subsumption architecture [Arkin, 1998] . We currently test and compare different ideas and proposals if they are appropriate for a scenario with the above mentioned properties of RoboCup-99.
Another problem closely connected to the mentioned one, is to find an appropriate mapping for a behavior between the sensor input and a useful response. The solutions proposed in the literature for this problem range from learned mappings to potential field methods [Latombe, 1991].

5 Summary and Discussion

This paper described the hardware and software architecture of our RoboCup-99 robot team. Our approach so far has been hardware oriented: we tried to find the most capable robot platform within our budget and tried to maximize the number, diversity and the quality of our sensors. To this end we are using sonars, IR sensors, a wide angle color camera, an omnidirectional camera, a compass and, if possible, a 2d scanning laser. Our underlying assumption is that at the current state of RoboCup play, improving the sensing capabilities will give a higher payoff than raising the speed of the robots or the onboard processing power or the “intelligence” of the robots. This is in contrast to the simulator or small size league, where all robots nearly have the same sensing capabilities. Our choice of sensors dictated the use of our pneumatic kicker and also the use of a larger PCI bus PC system with two frame grabbers. We use heavily specialized and optimized vision algorithms to keep the vision processing requirements low. The highlights of our software architecture are our method of sensor fusion which abstracts from individual sensors but keeps information about the reliability of sensor state with error data and the coupling of a reactive behaviour layer with additive behaviour outputs (rather than exclusive ones as in the subsumption architecture) with a planning component. We also believe we have found a good solution to update the global world model of each robot under unreliable radio ethernet communication. It will be seen wether we get to the point of "intelligent" team play at RoboCup-99 in Stockholm.

References

[Arkin, 1998] R. C. Arkin. *Behavior-based robotics.* MIT Press, 1998.

[Asada, 1998] M. Asada, editor. *RoboCup-98: Robot Soccer World Cup II, Proceedings of the second RoboCup Workshop*, 1998.

[Franz *et al.*, 1998] M. O. Franz, B. Schölkopf, H. A. Mallot, and H. H. Bülthoff. Where did i take that snapshot? scene-based homing by image matching. *Biol. Cybern.*, 1998.

[Gutmann *et al.*, 1998] J-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. The cs freiburg team. In *Proc. of the second RoboCup Workshop*, pages 451–459, 1998.

[Kitano *et al.*, 1997] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *Proc. of the first Int. Conf. on Autonomous Agents*, pages 340–347, 1997.

[Latombe, 1991] J.C. Latombe. *Robot Motion planning.* Kluwer Academic Publishers, 1991.

[Mojaev and Zell, 1998] A. Mojaev and A. Zell. Sonardaten-integration für autonome mobile roboter. In Levi P., Ahlers R.-J., May F., and Schanz M., editors, *Mustererkennung 1998*, pages 556–565. Springer-Verlag, 1998.

[Weiss and v. Puttkammer, 1995] G. Weiss and E. v. Puttkammer. A map based on laserscans without geometric interpretation. In U. Rembold et al., editors, *Intelligent Autonomous Systems*, pages 403–407. IOS Press, 1995.