Team Description of the GMD RoboCup-Team

A. Siegberg, A. Bredenfeld, H. Guenther, H.U. Kobialka, B. Klaassen, U. Licht, K.L. Paap, P.G. Ploeger, H. Streich, J. Vollmer, J. Wilberg, R. Worst, and T. Christaller

GMD,
Institute for System Design Technology,
D-53754 Sankt Augustin, Germany,
siegberg, bredenfeld, guenther, kobialka, klaassen, licht, paap, ploeger, streich, vollmer, wilberg, worst, christaller@gmd.de

Abstract. The article describes the structure of the GMD robots developed for the RoboCup '98. The hardware of these robots consists of an aluminum chassis with differential drives. They have low level sensors (odometry, distance sensors etc.) and a NewtonLab vision system. The software is organized in a layered structure using a uniform design pattern on each layer. A synchronous communication paradigm is adopted for the information exchange between the different layers.

1 Introduction

In order for a robot team to actually perform a soccer game [1] [3] various technologies must be incorporate: design principles of autonomous robots (hardware and software design), networking, strategy acquisition, sensor fusion and actor controlling. We present the design methodology for a team of middle sized robots.

The next section contains our design guidelines for the RoboCup scenario. Section 3 gives a detailed description of the GMD-robot. The system software is structured in four layers. Each layer uses a similar design pattern. Section 4 and 5 present the details of this structure. In section 6 we introduce our first strategy planning concepts. Concluding the paper we present results of our robotic activities.

2 Design Guidelines

For the RoboCup project we define some design guidelines to develop skillful soccer playing robots. The robots should cope with a dynamically changing environment and a quickly moving ball.

Autonomous systems, such as these robots, need to be physically strong, computational fast and behaviorally accurate to behave sensible in the rapidly changing environment of a soccer field. In our general architecture we implement therefore strong motors, simple sensors and simple stereotype movements to guarantee the basic qualifications.

M. Asada and H. Kitano (Eds.): RoboCup-98, LNAI 1604, pp. 473-478, 1999.

Since teamwork and individual skills are fundamental factors on the soccer playground the complexity of the design increases. The soccer robots must have the basic soccer skills, such as shooting, passing and recovering the ball from an opponent. The robot must be able to evaluate its position with respect to its teammates and opponents. The robot has to decide its next action while at the same time following the rules of the game [4]. Finally a communication structure is needed because this offers the robot the possibility to communicate with its teammates and an external PC.

In the next section we describe the first prototype of the GMD robot.

3 GMD Robots

The GMD-robots consist of the following components:

PC and Wavelan The PC on each robot is a small laptop and a wavelan ethernet card is used for the wireless communication.

Vision System Our vision system is a camera together with a board from NewtonLab, which is used to detect and track the ball. A servo motor is used to change the viewing direction of the camera. This allows an angular ball tracking range of approximately 270 degree. The soccer robot depends almost entirely on its visual input to perform its task.

Sensors 4 color detectors determine the color of the surrounding objects. For RoboCup each robot must distinguish the colors of the ball, goals, wall and other objects in the playground. After learning a very fast object recognition is possible. 16 gray scale sensors are placed around the robot's body. They are able to measure the light intensity in the immediate vicinity of the robot.

Touch sensors Bumper, constructed at the GMD, detects a collision of the robot e.g. with opponents or the wall. Furthermore the robot uses infra-red distance sensors.

Motor The motor speed is controlled by PWM (pulse width modulation). A special brake function is included.

Micro-Controller The robot is equipped with three 16-Bit Micro-Controllers [5] which are connected via a CAN-bus.

The total implementation of the robot was performed in different layers. The four major layers are described in the next section.

4 System structure

For a robot which has to play soccer a wide range of different components need to be integrated. Because of the complexity of the whole structure we divide it in four layers. On each layer we uses a uniform software design pattern (Fig. 1). A synchronous communication scheme is adopted for the communication among the layers (Fig. 2).

The server PC is an external PC which is able to communicate via a non-real-time LAN (local area network) with the Robot PC. The external robot

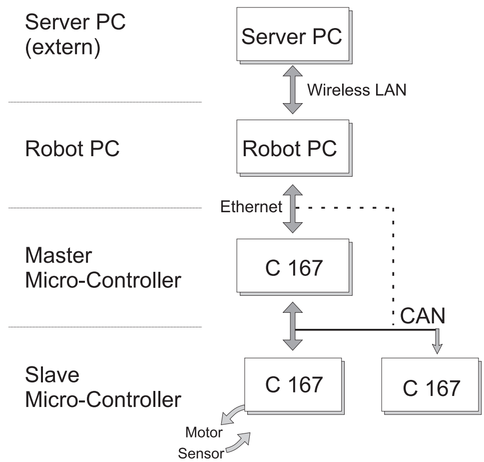

Fig. 1. The system structure with different layers of implementation

is used to exchange the position information among the robots. A star-type network topology is used. Each robot sends its local vision data (own position, ball position, and distance to other robots) to the server. The server merges the data to for a coherent data set which is broadcasted to the different Robot PCs.

The Robot PC is installed on top of each robot. On this PC a subset of local strategies is implemented because this makes each robot able to act in a useful autonomous way even when the network is down and the connection to the server PC is not active.

The master-controller and the slave-controllers define the real-time layers. There are three 16-Bit Micro-Controllers (C167). These controllers are connected via a CAN-bus. The master-controller performs the low level decision making. The first C167 slave-controller processes the motor-, odometry, color cell-, bumper- and line detector-data and it is responsible for the obstacle avoidance. The second C167 slave-controller processes the data of the distance- and gray-scale-sensors. The three micro-controllers perform the real-time processing of the robot system and communicate via the CAN bus whereas the server-PC and the robot-PC communicate via ethernet. If real-time processes are necessary in the whole system it is also possible to connect the controllers via the CAN bus directly to the robot-PC (this involves a changing of the robot-PC operating-system).

The whole communication and data transmission between the four layers is organized in a synchronization scheme with slot frames (Fig. 2).

The master-controller acts like a synchronous time clock and is responsible for the synchronization task between the different layers. The schedulers of each layer operate at different synchronization rates. The master controller runs at the highest synchronization rate. The rates of all other layers are derived from this rate. Schedulers on each layer are synchronized by the master controller.

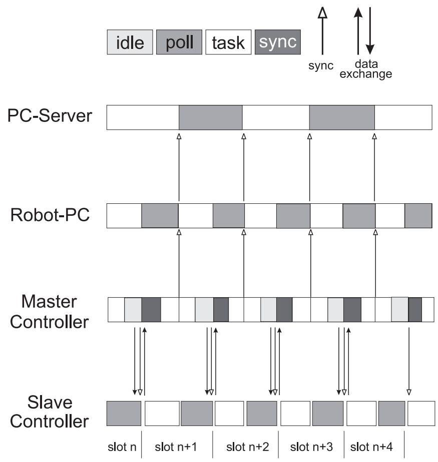

Fig. 2. Synchronization of the different layers with slots.

The conditions of the server-PC, the robot-PC and the slave-controller change between polling and task processing.

5 Software Design

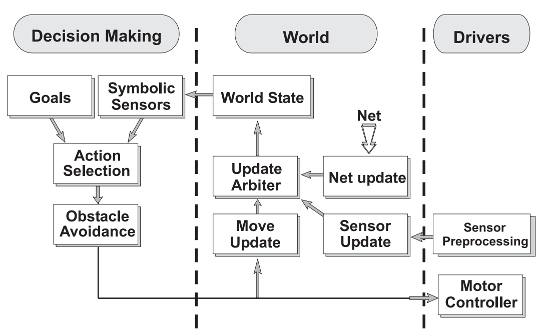

Fig. 3. The three most important parts of the software design.

A uniform software design pattern is used on each of the system layers. The pattern is depicted in Fig. 3. The software pattern is divided into three parts.

Drivers A sensor preprocessing is performed and the motor controller is implemented. The sensor preprocessing maps real world events to an internal representation.

World The world models the actual playground and the robot state depending on the sensor values and the world state received via the network.

Decision Making For the decision making part the values of the world state are reduced to symbolic sensors. The symbolic sensors and the robot goal are the input for the decision making process. After an action is selected a check for a collision is made. If there is no collision, the action command is transmitted to the motor controller.

A detailed description of our first strategies are described in the next section.

6 Strategy Planning

Each robot must react appropriately to different situations on the soccer field. The robot needs information about the environment and itself. This information delivers the world. The world presents the actual states of the sensors.

For the RoboCup competition in 1998 we implemented a set of strategies. These strategies are the basic skills for the robot to act on the playground.

We divided the strategy in two parts (1) actions in the opponent half and (2) actions in the own half. Important values for the robot are the X- and Y-coordinates of itself, his team members, the opponents and the ball. These values are transmitted to each robot via the world interface.

The first priority for our robot in the opponent half is the ball-goal line (Fig. 4). When the robot detects the ball, it drives behind the robot on the ball-goal line. Next, it turns into the direction of the goal and shoots the ball to the goal.

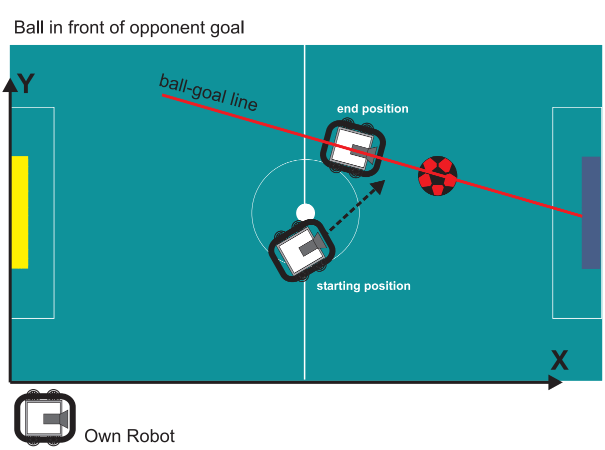

Fig. 4. Robot on the soccer field.

The first priority for the robot in its own half is the protection of the own goal. Therefore it always drives behind the ball. In this situation the robot ignores the ball-goal line. If the robot detects the opponent *and* the ball, it has the possibility to check the value of delta x. Delta x delivers the robot the distance between ball and itself. When the value is large it can directly drive on the ball-goal line. When the value of delta x is small the robot has to perform an obstacle avoidance.

7 Results and Future Work

Building a robot to play soccer is a big challenge in different fields. The range of technologies spans AI, robotic research and embedded system design. What we have learned after the RoboCup 1998 in Paris is the great importance of the sensors. In Paris we used mainly the vision sensor. But especially the vision sensor is a very complex one. We must take into account the movement of the sensor itself (because it is installed on a servo), the movement of our robots, and the movement of the ball.

In our system of the Paris RoboCup we tried to derive the movement operators directly from the vision data, i.e., estimate the ball distance (e.g., 20cm) and generate a movement command (e.g., `move 20`). This leads to oscillations and unstable behavior (in particular, when using fast turns). Thus only slow robot movements can be realized in this way. The situation will be remedied by using a behavior-oriented robot control [2].

The GMD-robots are developed for a spectrum of different missions. Of course for the RoboCup they will act as soccer players but our research goal is the implementation of a flexible and changeable platform. Researchers from different fields can use the robot platform to tackle problems like real-time sensor fusion, reactive behavior, strategy acquisition, learning, real-time planning, multi-agent systems, context recognition, vision, strategic decision making and intelligent motor control.

References

[1] S. Hedberg. Robots playing soccer? RoboCup poses a new set of AI research challenges. *IEEE Expert Intelligent Systems and Their Applications*, pages 5–9, September/October 1997.

[2] H. Jäger and T. Christaller. Dual dynamics: Designing behavior systems for autonomous robots. In *Proceedings of International Symposium on Artificial Life and Robotics*, pages 76–79, 1997.

[3] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. RoboCup – a challenge problem for AI. *AI Magazine*, 18(1):73–85, Spring 1997.

[4] RoboCup. Regulations & rules. http:// www.robocup.org /regulations /4.html, 1998.

[5] Siemens AG, Munich, Germany. *C167 Derivatives – 16-Bit CMOS Single-Chip Microcontrollers*, 3.96 edition, 1996.