FU-Fighters 2001 (Global Vision)

Raúl Rojas, Sven Behnke, Achim Liers, and Lars Knipping

Free University of Berlin, Institute of Computer Science
Takustr. 9, 14195 Berlin, Germany
{rojas|behnke|liers|knipping}@inf.fu-berlin.de
http://www.fu-fighters.de

1 Introduction

Our F180 team, the FU-Fighters, participated for the third time at the RoboCup competition. This year we used a heterogeneous team, consisting of improved differential drive robots and new omnidirectional robots. We designed new electronics and added prediction and path planning to the behavior control. Our team won fourth place in the SmallSize league competition.

2 Team Development

Team Leader: Raúl Rojas (college professor)
Team Members:
Sven Behnke (scientific staff): general design
Achim Liers (scientific staff): electronics, mechanics
Lars Knipping (scientific staff): behavior control
Bernhard Frötschl (scientific staff): web site, organization
Mark Simon (student): global vision, user interface
Kirill Koulechov (student): behavior control, microcontroller programming
Lars Wolter (student): behavior control, user interface
Oliver Tenchio (student): mechanics

3 Mechanical and Electrical Design

For RoboCup 2001 we built a new generation of omnidirectional robots as shown in Figure 1. The robots are equipped with three DC-motors that have an integrated 19:1 gear and an impulse generator with 16 ticks per revolution. They drive special wheels which allow for omnidirectional movement [4, 5]. Further, the robots use a rotating kicking device. We also used specialized robots as defenders or for offense when appropriate. They are also shown in the figure.

For local control we developed a new microcontroller board that is based on the Motorola HC12, as shown in Figure 2. This controller features 8KB RAM, 2KB EEPROM, 128KB flash, several timers, four PWM-units, digital I/Os, eight analog inputs, two RS-232 serial lines, and a CAN interface. The board can drive with pulse-width modulation four DC-motors and captures the impulses

Fig. 1. Different robots designs. From left to right: classical two wheeled design, two wheeled defender, and robot with omnidirectional drive.

generated by them. Further, four servos can be steered and eight on/off switches can be used. A mezzanine board contains a radio transceiver SE200 working in the 433MHz band that can be tuned to 15 channels in 100kHz steps.

The robots receive commands via a wireless serial link with a speed of 19,200 baud. The host sends 10-byte packets that include address, control bits, motor speeds, and checksum. The microcontroller decodes the packets, checks their integrity, and sets the target values for the control of the motor speeds.

Our robots are powered by 8 Ni-MH rechargeable mignon batteries. We implemented locally a PID-control of the motor speeds.

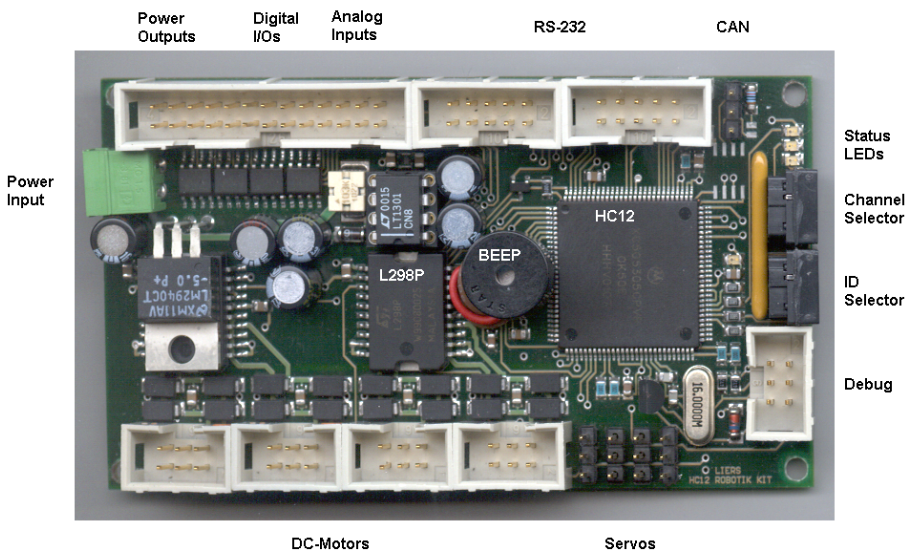

Fig. 2. HC12 microcontroller board.

4 Computer Vision and Prediction

The only physical sensor for our behavior control software is a S-VHS camera that looks at the field from above and outputs a video stream in NTSC format that is captured by a PC. The computer vision system running on the PC finds and tracks the ball and the robots. We used an improved version of the system described in [2]. Unfortunately, the information about the world is extracted with a delay caused by communication, mechanical constraints, and the computer vision system. The feedback (the result) of an action decision is perceived typically between 100ms and 150ms after the decision has been made. This causes problems when robots move fast, producing overshooting or oscillations in movement control. One way to deal with the delay would be to move more slowly, but this is often not desirable.

Our approach to solve that problem is to predict the movement of the robots for the next few frames. We feed the last robot positions and orientations (relative to the current robot position and orientation) and action commands to a feed forward neural network that is trained to predict the robot position and orientation for a point in time 130ms away. We train the network with recorded data before the game. The predicted positions and orientations are used for behavior control. This approximately cancels the effects of the delay and allows for fast and exact movement control.

5 Hierarchical Reactive Behavior and Path Planning

We use a hierarchy of reactive behaviors to control the robots. Simple behaviors are arranged in layers that work on different time scales. Fast primitive behaviors, such as taxis are implemented in the lowest layer. More complex, but slower behaviors are produced by the higher layers of the system. A more detailed description of our control architecture is given in [1, 3].

Since the field is very crowded and significant contact with other robots must be avoided, we implemented a path planner on the second layer of the behavior control system. The path planner finds a path from a start point (the current robot position) to a goal (the desired robot position). Using dynamic programming and best-first search it computes the cheapest path on a grid that avoids obstacles such as other robots or the defense area. Figure 3 illustrates the behavior of the path planner. After the path has been found, the first point A where the path significantly turns is determined and communicated as target position to the lowest control level. As the robot moves towards A, the path is constantly re-planned and the turn point A moves towards the goal.

6 Future Work

In order to reduce the control delay, we plan to implement a larger portion of our behavior hierarchy directly on the robots. To allow for faster and more exact movement, fast local sensors for robot motion and obstacles are needed. We also

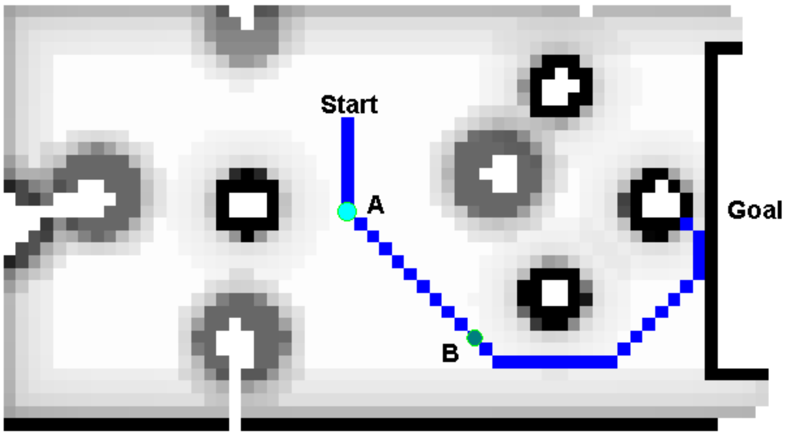

Fig. 3. Pathplanner. The shading of cells corresponds to cost. Expensive dark cells are caused by obstacles, such as other robots and walls, and by the defense area that should not be visited by field players. The dark blue cells show the path with the smallest aggregated costs that is computed by a best first search that visits only parts of the grid. White cells need not be considered. Point A is the first significant turn on the path. This point is communicated to the lowest layer in behavior control together with the direction towards point B. The robot drives from start towards A such that it can next turn to B. As the robot approaches the goal, the path is updated and points A and B move closer to the goal.

plan to develop more sophisticated ball handling mechanisms and controlled kicks that are needed for complex behaviors such as passing.

References

1. Behnke, S., Frötschl, B., Rojas, R., Ackers, P., Lindstrot, W., de Melo, M., Schebesch, A., Simon, M., Sprengel, M, Tenchio, M.: Using hierarchical dynamical systems to control reactive behavior. *In: Veloso, M., Pagello, E., Kitano, H. (eds): RoboCup-99: Robot Soccer World Cup III*, pp. 186-195, Springer, 2000.
2. Simon, M., Behnke, S., Rojas, R.: Robust Real Time Color Tracking. *In: Stone, P., Balch, T., Kraetszchmar (eds): RoboCup-2000: Robot Soccer World Cup IV*, pp. 239-248, Springer, 2001.
3. Behnke, S., Rojas, R.: A Hierarchy of Reactive Behaviors Handles Complexity. *In: Hannebauer, M., Wendler, J., Pagello, E. (eds.): Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, pp. 125-136, Springer, 2001.
4. Reshko, G., Mason, M., Nourbakhsh, I.: Rapid Prototyping of Small Robots, *Technical Report*, CMU, 2000.
5. Palm Pilot Robot Kit. The Robotics Institute, CMU, http://www.cmu.edu/∼pprk.