FU-Fighters 2000

Raúl Rojas, Sven Behnke, Lars Knipping, and Bernhard Frötschl

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

1 Introduction

Our F180 team, the FU-Fighters, participated for the second time at the RoboCup Competition. Our main design goal for RoboCup'2000 was to build a heterogeneous team of robots with different shapes and characteristics, and to use new kicking devices. Different sets of robots can be selected to adapt to the strategies of other teams. Our team finished second, as the runner-up to Big Red.

2 Team Development

Team Leader: Raúl Rojas (college professor)
Team Members:
Sven Behnke (scientific staff): general design
Lars Knipping (scientific staff): behavior control
Bernhard Frötschl (scientific staff): web site, mechanics
Achim Liers (scientific staff, did not attend competition): electronics
Wolf Lindstrot (student): behavior control
Mark Simon (student): computer vision, user interface
Kirill Koulechov (student): behavior control, communication
Lars Wolter (student): behavior control, user interface
Oliver Tenchio (student): mechanics, electronics

3 Mechanical and Electrical Design

We built three types of field players and goal keepers as shown in Figure 1. All robots have sturdy aluminum frames that protect the sensitive inner parts. They have a differential drive with two active wheels in the middle and are supported by two passive contact points. Two Faulhaber DC-motors allow for a maximum speed of about 1.5-2.5 m/s. The motors have an integrated 19:1 gear and an impulse generator with 16 ticks per revolution. One distinctive feature of some of our robots is a kicking device that consists of a rotating plate that can accumulate the kinetic energy produced by a small motor and release it to the ball on contact. Our smaller robots use a kicking device that is based on solenoids that can be activated faster than the rotating plate.

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 547-550, 2001.

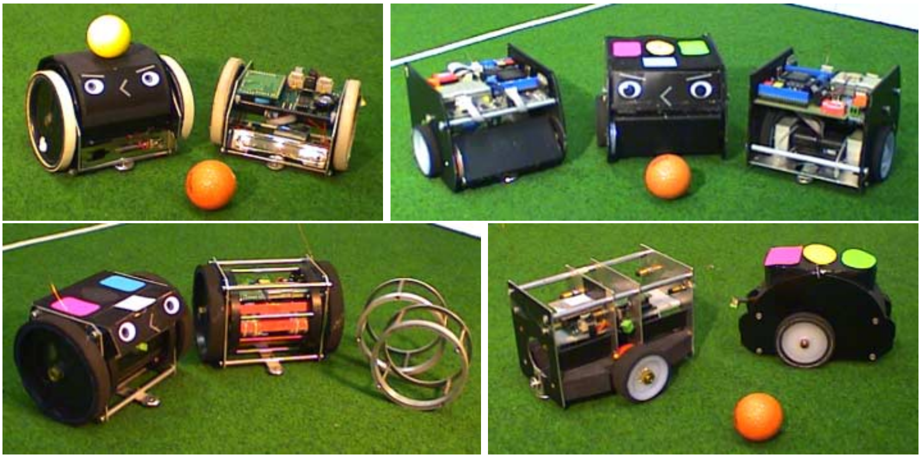

Fig. 1. Different generations of robots.

For local control we use C-Control units from Conrad electronics. They include a microcontroller Motorola HC05 running at 8 MHz with 8 KB EEPROM for program storage, two pulse-length modulated outputs for motor control, a RS-232 serial interface, a free running counter with timer, analog inputs, and digital I/O. The units are attached to a custom board containing a stabilized power supply, a dual-H-bridge motor driver L298, and 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 8-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. No attempt is made to correct transmission errors, since the packets are sent redundantly.

Our robots are powered by 8 Ni-MH rechargeable mignon batteries. To be independent from the charging state of the batteries, we implemented locally a closed loop control of the motor speeds.

4 Tracking Colored Objects in the Video Input

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. Using a PCI-framegrabber we feed the images into a PC. We capture RGB-images of size 640×480 at a rate of 30 fps and interpret them to extract the relevant information about the world. Since the ball and the robots are color-coded, we designed our vision software to find and track multiple colored objects. These objects are the orange ball and the robots marked with two colored dots in addition to the yellow or blue team ball.

To track the objects we predict their positions in the next frame and then inspect the video image first at a small window centered around the predicted position. We use an adaptive saturation threshold and intensity thresholds to separate the objects from the background. The window size is increased and larger portions of the image are investigated only if an object is not found.

The decision whether or not the object is present is made on the basis of a quality measure that takes into account the hue and size distances to the model and geometrical plausibility. When we find the desired objects, we adapt our model of the world using the estimates for position color, and size. We also added an identification module to the vision system that recognizes the robots by looking at a black and white binary code.

5 Hierarchical Generation of Reactive Behavior

The Dual Dynamics control architecture, proposed by Jäger [3], describes reactive behaviors in a hierarchy of control processes. Each layer of the system is partitioned into two modules: the activation dynamics that determines whether or not a behavior tries to influence actuators, and the target dynamics, that determines strength and direction of that influence. The different levels of the hierarchy correspond to different time scales. The higher level behaviors configure the lower level control loops via activation factors that determine the mode in which the primitive behaviors are. These can produce qualitatively different reactions if the agent encounters the same stimulus again, but has changed its mode due to stimuli that it saw in the meantime.

A more detailed description of our control architecture, that is based on these ideas, is given in [1]. The robots are controlled in closed loops that use different time scales. We extend the Dual Dynamics scheme by introducing a third dynamics, the perceptual dynamics. Here, either slow changing physical sensors are plugged in at higher levels, or readings of fast changing sensors, like the ball position, are aggregated to slower and longer lasting percepts. Since we use temporal subsampling, we can afford to implement an increasing number of sensors, behaviors and actuators in the higher layers.

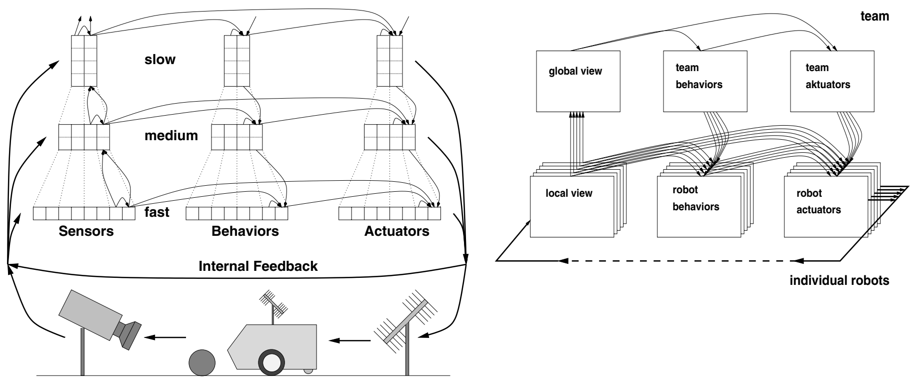

Fig. 2. Reactive control architecture.

Behaviors are constructed bottom up: First, processes that react quickly to fast stimuli are designed. Their critical parameters, e.g. a target position, are determined. When the fast behaviors work reliably, the next level can be added. This level can now influence the environment either directly by moving slow actuators or indirectly by changing critical parameters in the lower level.

In the lowest level of the field player only two behaviors are implemented. These are a parameterized taxis behavior and an obstacle avoidance behavior.

On the next level various behaviors use them. They approach the ball, dribble with the ball, kick the ball, free the ball from corners, home, and so on. Here, we also implemented a ball prediction that allows for anticipative actions. On the third level, we dynamically adjust the home positions of the players, such that they block opponent players or position themselves freely to receive passes.

Each of our robots is controlled autonomously from the lower levels of the hierarchy using a local view to the world. For instance, we present the angle and the distance to the ball and the nearest obstacle to each agent. In the upper layers of the control system the focus changes. Now we regard the team as the individual. It has a slow changing global view to the playground and coordinates the robots as its extremities to reach strategic goals.

In the first team level, we decide which robot should take the initiative and go for the ball. The remaining players are assigned to supporting roles. Passing and strategy changes would be implemented in higher team layers.

6 Future work

Our goal is to make the robots more autonomous. Therefore, we want to add a local omni-directional camera to the robots. Initially we want to transmit the images to an external computer, where they can be analyzed. Later, the vision algorithms should be ported to an on-board computer. We also plan to add other local sensors that can measure the robot's movement. One further step towards autonomy will be to implement the behavior control locally.

The other research direction, we are interested in, is learning. We plan to use reinforcement learning to adapt the parameters of the system. We also hope to improve team play to allow for an increased number of players on a larger field.

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, 186–195, Springer, 2000.
2. Christaller, T.: Cognitive Robotics: A New Approach to Artificial Intelligence. In: Artificial Life and Robotics, Springer, 3/1999.
3. Jäger, H., Christaller, T.: Dual Dynamics: Designing Behavior Systems for Autonomous Robots. In: Fujimura, S., Sugisaka, M. (eds:) Proceedings International Symposium on Artificial Life and Robotics (AROB '97), Beppu, Japan, 76–79, 1997.