RoboCup'99 (F180) Team Description: FU-Fighters

FU-Fighters

Sven Behnke, Bernhard Frötschl, Raul Rojas, Peter Ackers, Manuel de Melo, Mark Preier, Andreas Schebesch, Mark Simon, Martin Sprengel, Oliver Tenchio

SB, BF, RR, PA, MD, MP, AS, MS, MS, OT: Free University of Berlin, Institute of Computer Science
“’

Abstract. *This paper describes the team FU-Fighters that is qualified for the RoboCup'99 F180-league competition. After presenting the mechanical and electrical design of our robots, the paper explains the hierarchical control architecture which generates the behavior of individual agents and the team. This reactive approach is mainly based on the Dual Dynamics concept developed by H. Jäger. In addition, the paper describes how we solved the problems of vision and radio communication.*

1 Introduction

Our group is a first time participant in the RoboCup competition. We became interested in RoboCup in spring '98. To prepare for the competition, we first analyzed the different approaches of other teams in a seminar. In autumn '98 we started to construct robots for the F180 league and to design software for vision, control, and communication. Now, the main components are working and we are using the remaining time to refine the generated behaviors. Our main motivation to participate in RoboCup is to use it as a platform for developing approaches for problems like vision and control using neural networks and learning [2].

2 Mechanical Design

Our robots are designed in compliance with the new F180 size regulations. We built four identical field players and a goal keeper. All robots have stable aluminum frames that protect the sensitive inner parts.

They have a differential drive with two active wheels in the middle and are supported by one or two passive spheres that can rotate in any direction. Two Faulhaber DC-motors allow for a maximum speed of about 1 m/s. The motors have an integrated 19:1 gear and an impulse generator with 16 ticks per revolution.

3 Electrical Design

For local control we use C-Control units from Conrad electronics. They include a Motorola microcontroller HC05 running at 4 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 functions, 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 from Radio Communication Systems working in the 433 MHz band. The robots are powered by 8 + 4 Ni-MH rechargeable mignon batteries.

4 Video

The only physical sensor for our control software is a S-VHS camera that looks from above at the playground and outputs a video stream in NTSC format. Using a PCI-framegrabber we input the images into a PC running MS-Windows. We capture RGB-images of size 640x480 at a rate of 30 fps and interpret them to extract the relevant information about the world. Since the ball as well as 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 that have been marked with 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 measured parameters, such as position, color, and size.

For each of our robots we compute a local view that is used as input for the control software.

5 Behavior

In 1992, the programming language PDL was developed by Steels and Vertommen for the stimulus driven control of autonomous agents [5]. This language has been used by a number of groups working in behavior oriented robotics [4]. It allows the description of parallel processes that react to sensor readings by influencing the actuators. Many primitive behaviors, like

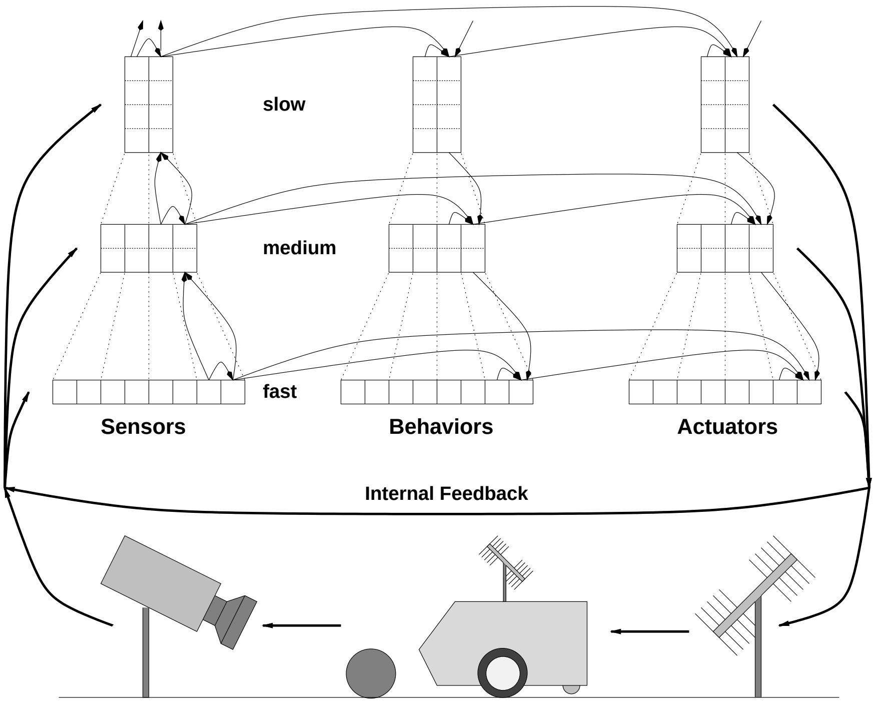

Figure 1: Sketch of the control architecture.

taxis, are easily formulated in such a framework. On the other hand, it is difficult to implement more complex behaviors in PDL that need information about slow changes in the environment.

The Dual Dynamics control architecture, developed by Herbert 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.

Our control architecture is based on these ideas, as shown in Figure 1. A more detailed description 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, namely the perceptual dynamics shown on the left side. Here, either slow changing physical sensors are plugged in at the higher levels, or the readings of fast changing sensors, like the ball position, are aggregated by dynamic processes to slower and longer lasting percepts. Since we use a subsampling in time, we can afford to implement an increasing number of sensors, behaviors and actuators in the higher layers without an explosion of computational costs.

The behaviors are constructed in a bottom up fashion: First, the processes that should react quickly to fast changing stimuli are designed. Their critical parameters, e.g. a mode parameter or a target position, are determined.

When the fast processes work reliably with constant parameters, the next level can be added to the system. This level can now influence the environment either directly by moving slow actuators or indirectly by changing the critical parameters of the processes in the lower level.

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.

6 Communication

The actions that have been determined by the control module are transmitted to the robots via a serial communication line with a speed of 9600 baud. We use radio transmitters operating on a single frequency that can be chosen between 433.0 MHz and 434.5 MHz in 100 KHz steps. The host sends commands in 8-byte packets that include address, control bits, motor speeds, and checksum. A priority value can be used to direct more packets to the most active players.

The microcontroller on the robots 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.

To be independent from the charging state of the batteries, we implemented a closed loop control of the motor speeds. 122 times per second the microcontroller counts the impulses from the motors, computes the differences to the target values and adjusts the pulse length ratio for the motor drivers accordingly.

7 Summary

We designed robust and fast robots with reliable radio communication and high speed vision. To generate actions, we implemented a reactive control architecture with interacting behaviors on different time scales.

This should allow us to make a contribution to the RoboCup'99 competition.

Acknowledgments

We thank the companies Conrad ELECTRONICS GmbH, Dr. Fritz Faulhaber GmbH & Co KG, and Siemens ElectroCom Postautomation GmbH for their support that made this research possible.

References

[1] S. Behnke, B. Frötschl, R. Rojas, P. Ackers, W. Lindstrot, M. de Melo, M. Preier, A. Schebesch, M Simon, M. Sprengel, and O. Tenchio. Using hierarchical dynamical systems to control reactive behavior. In *Proceedings RoboCup Workshop at the International Joint Conference on Artificial Intelligence (IJCAI '99) – Stockholm (accepted)*, 1999.

[2] T. Christaller. Cognitive robotics: A new approach to artificial intelligence. *Artificial Life and Robotics*, (3), 1999.

[3] H. Jäger and T. Christaller. Dual dynamics: Designing behavior systems for autonomous robots. In S. Fujimura and M. Sugisaka, editors, *Proceedings International Symposium on Artificial Life and Robotics (AROB '97) – Beppu, Japan*, pages 76–79, 1997.

[4] E. Schlottmann, D. Spenneberg, M. Pauer, T. Christaller, and K. Dautenhahn. A modular design approach towards behaviour oriented robotics. Arbeitspapier 1088, GMD, 1997.

[5] L. Steels. The pdl reference manual. AI Lab Memo 92-5, VUB Brussels, 1992.