

The IUB 2002 Rescue Robot Team

Andreas Birk, Holger Kenn, Martijn Rooker, Agrawal Akhil, Balan Horia Vlad, Burger Nina, Burger-Scheidlin Christoph, Devanathan Vinod, Erhan Dumitru, Hepes Ioan, Jain Aakash, Jain Premvir, Liebold Benjamin, Luksys Gediminas, Marisano James, Pfeil Andreas, Pflingsthorn Max, Sojakova Kristina, Suwanketnikom Jormquan and Wucherpennig Julian

International University Bremen, Germany
a.birk@iu-bremen.de, <http://www.iu-bremen.de/RoboCup/>

1 Introduction

The main research topics for the IUB rescue team are twofold. First, we work on the systems engineering side to develop robust and versatile robots that are nevertheless rather inexpensive. The low cost factor should allow us to use larger groups of robots in scenarios with a high risk of losing individual systems. Second, semi-autonomy is investigated to allow a safe and efficient integration of teleoperation via a human operated basestation and various autonomous behaviors on board of the robots. The IUB rescue robot team is also used for education, as it is a regular part of the undergraduate program in Electrical Engineering and Computer Science (EECS) at IUB [RCI02].

2 The Hardware Side

The implementation of the rescue robots is based on the so-called CubeSystem, a kind of construction kit for robotic systems. The CubeSystem is used in basic and applied research, industrial projects and academic education (see e.g. [BWBK99,BWB⁺98,BB98,Bir98]).

The center of the CubeSystem is the so-called RoboCube controller hardware (figure 1) based on the MC68332 processor. The RoboCube has an open bus architecture which allows to add “infinitely” many sensor/motor-interfaces at the price of bandwidth. The RoboCube’s basic set of ports consists of 24 analog/digital (A/D) converter, 6 digital/analog (D/A) converter, 16 binary Input/Output (binI/O), 5 binary Inputs, 7 timer channels (TPC), and 3 DC-motor controller with quadrature-encoding (QDEC). The RoboCube is described in more detail in [BKW00,BKW98].

In addition to its central component, the RoboCube as controller hardware, the CubeSystem provides additional hardware, including electronics and mechanics, and software components. For the more challenging locomotion tasks that are needed for rescue robots, a new base was developed that features six actively driven wheels (figure 1). The CubeSystem also features a special operating system, the CubeOS [Ken00], providing component based support for realtime and control functions.

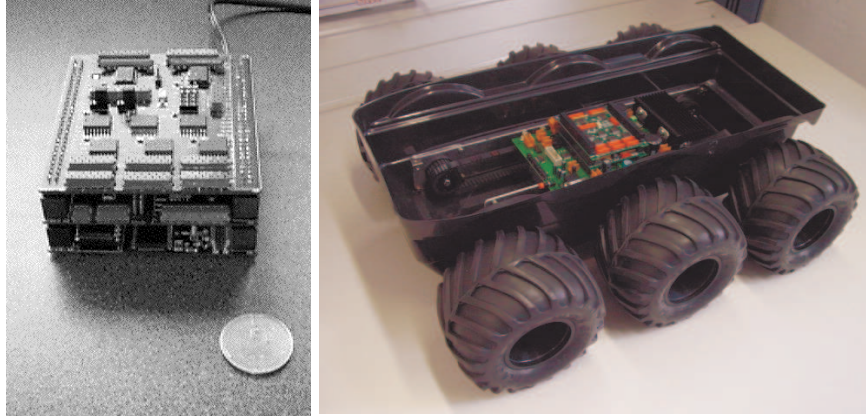


Fig. 1. Left: The RoboCube, an extremely compact embedded computer for robot control. Right: The rescue mobile base with six actively driven wheels. It is completely constructed from CubeSystem components including the RoboCube as controller, the motor- and sensor-modules, as well as the battery-management hardware.


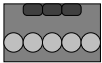

cameras		mobile PC 4x USB-cameras video compression WaveLAN RF-ethernet
sensor moduls		CubeSystem 5x Ultrasound Sonar 6x Active Infrared optional (Pyro, Temp., Smoke)
mobile base		Odometry and Positioning Motioncontrol Motorcontrol Battery- and Powermanagement

Fig. 2. A schematic overview of the different components of a rescue robot.

On the rescue robots, a mobile PC is used to service RF-ethernet and to compute video-compression. All control and service related data going to and coming from the cockpit is directly relayed from the RF-connection to the RoboCube which handles all service and control related tasks on the rescue robot.

3 The Software Side

The IUB rescue robots are connected via standard networking technology, especially RF-ethernet, to their cockpit(s) from where they can be tele-operated

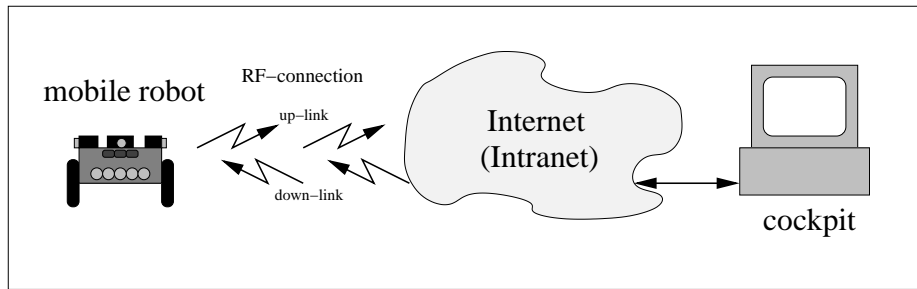


Fig. 3. The IUB rescue robots are teleoperated via standard networking technology from a so-called cockpit by a human operator. Despite the human in the loop, each robot needs quite some autonomous functionality ensuring its proper behavior as the network performance is unknown and can even break completely down, especially as wireless components are involved.

(figure 3). The main load is on the up-link in form of streams of video and other essential data, whereas only a few densely encoded highlevel motion commands are sent down-link to the base.

Due to the limitations of wireless connections and the complexity of rescue operations, the full operation of a robot can not be constantly supervised by a human operator, i.e., the robots have to be semi-autonomous. So, the commands by the human operator that are high-level and maybe already outdated due to a broken RF-link have to be supplemented by various autonomous behaviors on board of the robots. In doing so, there are two major issues, namely ensuring fail-safe-guarantees (FSG) and quality-of-service (QoS). FSG must never be violated at any cost. This means for example that major obstacles and gaps in the ground must be avoided or that the base must be stopped to avoid serious damages. QoS in contrast defines constraints which maximize utility as long as they are not violated. A timely response to requests from the operator for example ensures that the mobile robot moves along its path as desired. If these constraints are occasionally violated, they should at most cause some slight inconveniences to the operator, but they never must put the whole device or mission at risk. A detailed description of this software architecture can be found in this volume [BK02].

4 Conclusion

The paper gave a short introduction to the IUB 2002 Rescue Robot Team that consists of senior researchers and undergraduate students. The research of the team focus on two aspects. First, it is attempted on the systems engineering side to develop robust and versatile bases that are nevertheless not costly to allow the usage of larger groups of robots in situations where there is a high risk of losing some bases. In doing so, the so-called CubeSystem is used that

centers around a special embedded controller, the RoboCube. Second, semi-autonomy is investigated to allow a safe and efficient integration of tele-operation via a human operated basestation and various autonomous behaviors on board of the robots. For this purpose a special control architecture was developed. This control architecture combines hard realtime control on the bases with quality of service for the human operator at the base station. In respect to education, the team is used as a regular part of the undergraduate program in Electrical Engineering and Computer Science (EECS) at IUB.

References

- [BB98] Andreas Birk and Tony Belpaeme. A multi-agent-system based on heterogeneous robots. In Alexis Drogoul, Milind Tambe, and Toshio Fukuda, editors, *Collective Robotics, CRW'98*, LNAI 1456. Springer, 1998.
- [Bir98] Andreas Birk. Robot learning and self-sufficiency: What the energy-level can tell us about a robot's performance. In *Proceedings of the Sixth European Workshop on Learning Robots*, LNAI 1545. Springer, 1998.
- [BK02] Andreas Birk and Holger Kenn. A control architecture for a rescue robot ensuring safe semi-autonomous operation. In Gal Kaminka, Pedro U. Lima, and Raul Rojas, editors, *RoboCup-02: Robot Soccer World Cup VI*, LNAI. Springer, 2002.
- [BKW98] Andreas Birk, Holger Kenn, and Thomas Walle. Robocube: an "universal" "special-purpose" hardware for the robocup small robots league. In *4th International Symposium on Distributed Autonomous Robotic Systems*. Springer, 1998.
- [BKW00] Andreas Birk, Holger Kenn, and Thomas Walle. On-board control in the robocup small robots league. *Advanced Robotics Journal*, 14(1):27 – 36, 2000.
- [BWB⁺98] Andreas Birk, Thomas Walle, Tony Belpaeme, Johan Parent, Tom De Vlaminck, and Holger Kenn. The small league robocup team of the vub ai-lab. In *Proc. of The Second International Workshop on RoboCup*. Springer, 1998.
- [BWBK99] Andreas Birk, Thomas Walle, Tony Belpaeme, and Holger Kenn. The vub ai-lab robocup'99 small league team. In *Proc. of the Third RoboCup*. Springer, 1999.
- [Ken00] Holger Kenn. Cubeos, the manual. Technical Report MEMO 00-04, Vrije Universiteit Brussel, AI-Laboratory, 2000.
- [RCI02] The iub robocup team website. <http://www.iu-bremen.de/RoboCup>, 2002.