RoboCup Rescue 2017 Team Description Paper Rescube

Péter Kopiás and Márton Krauter

Info

Team Name:	Rescube
Team Institution:	Datastart Kft.
Team Leader:	Péter Kopiás
Team URL:	http://rescube.hu/

RoboCup Rescue 2017 TDP collection:
http://wiki.robocup.org/Robot_League

***Abstract*—The Team Description Paper of team Rescube gives a comprehensive overview of the team background, its robots and infrastructure in development for the 2017 RoboCup Championship in Nagoya, Japan. Based on the experiences of the previous competitions, team Rescube has further improved its versatile four wheel driven robot with a robotic arm with the ability to effectively meet the skills required at Rescue competitions. It features a special wheel design with variable geometry and an extensible "giraffe-neck" that affords great flexibility.**

***Index Terms*—RoboCup Rescue, Team Description Paper, Rescube.**

I. Introduction

TEAM Rescube has won the Best Outdoor CarryBot Award, finished 9th overall in the Major Rescue League, and won the Robocup Design Award Sponsored By Flower Robotics at the RoboCup WorldCup 2016 Leipzig, Germany.

The members of team Rescube are young robotics enthusiasts from Hungary who share a challenge seeking attitude and willingness to hone their problem solving skills in common. With roots dating back to the 2000s the idea of Team Rescube was ignited by the RoboCup World Cup 2013 event and the team made its debut at the 2015 RoboCup German Open and successfully demonstrated its abilities in the RoboCup WorldCup 2016 in Leipzig.

A. Team Background

While most team members are mechanical, electrical or software engineers we also have numerous university students of diverse areas. We are self motivated and run the project by solely relying on personal and sponsorship budget in parallel to maintaining strong partnership with top Hungarian scientific and technology universities.

Our efforts in RoboCup Rescue Robot League are pioneering to raise awareness of the Hungarian academic and industrial sector to the matters of the young engineers. Besides trying to establish an environment where the engineering students could work on interesting challenges, our intention is to encourage the next generation of students in choosing scientific and engineering fields. We truly believe that with continuous efforts we could create a great balance of tasks, knowledge and resources to let the creative ideas grow into solutions. We volunteer to help in organizing Hungarian robotics competitions, and to hold presentations to motivate the youngest to work hard and achieve their goals.

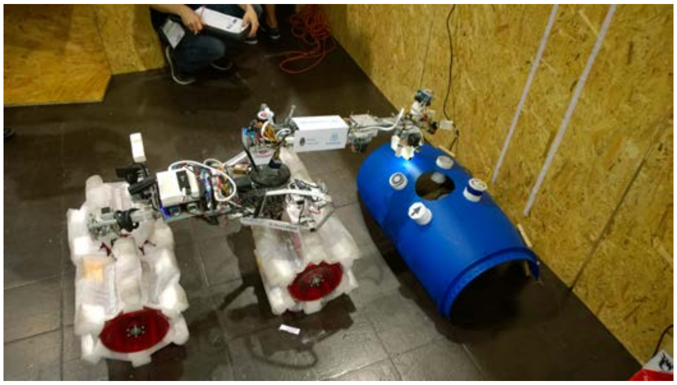

Fig. 1. Rescube R16 robot examining a possible victim

Team Rescube has presented it's robots and achievements in a series of scientific exhibitions Hungary-wide to inspire children and youngsters interested in the technology with great success.

We strive to maintain an open community characterized by tolerance, respect for the individual and a minimum of hierarchy. Last but not least we believe having fun together plays an indispensable role in any successful teamwork.

B. Improvements over Previous Contributions

1) Team conclusions: As this is a non-profit, voluntary hobby of the members, the resources available depend on members personal background and obviously on their morale.

After the previous competitions and exhibitions the team has gathered routine in the preparation, logistics and event management. Nonetheless we are constantly seeking more efficient technical processes and personal communications. As the team matures we are working hard on recruiting new team members and quickly integrating them into the team.

The lack of financial background is still the weakest point so serious efforts are required to keep the team running and to let us enter the different competitions. Our extended PR activity has resulted in an increased interest from people and companies, however this has no direct relation to our sponsorship opportunities.

Péter Kopiás is with the University of Óbuda, Budapest, Hungary, e-mail: kopias.peter@rescube.hu.

After the RoboCup Championship 2016 the team members were interviewed to gather and summarize the lessons learned and the take-aways to synthesize actions to be taken for a more successful next championship.

The key areas revealed (but are not limited to):

- importance/lack of practice and time management
- clear definition and communication of who-does-what-when
- improving handling of inevitable team fluctuation (knowledge transfer)
- lack of expertise in specific practice areas
- suboptimal logistics

We are actively seeking for the opportunities to improve and to widen the major and minor bottlenecks identified.

2) Software issues: Although using ROS as a software framework is an obvious choice, its learning curve (respectively that of the myriad possibly useful packages of varying level of documentation) is steep, and we had to distinguish between the software modules we spend time/energy on. Based on the experiences of the previous competitions it was clear that a unified software development environment is vital for the effective teamwork. We have put serious efforts to create a virtual development environment (using Vagrant and Ansible for configuration management) available to all team members at any time. As a result now we can have multiple virtual machines servers (also for the operator and robots computers) while all the configuration data is declarative and are contained in a version control system (git). The team-wide use of this development environment enables the new members to start contributing faster, use machines of each other's, and also helps keeping the software stack synchronised and centralises the management tasks to the senior members hands thus reducing maintenance costs. We have made plans to release the devops toolchain to the community, when it reaches the appropriate level of maturity.

Since 2015 we have learned that although autonomy is a great scientific feature in a real rescue mission, however the efficient human-robot interface is more important so we have decided to focus our efforts on lowering the operator mental load with helper tools and automated decision making.

3) Experimental VR user interface: We have learned that driving the robots through cameras is a very challenging task for the operator even if we have enough cameras located at every parts of the robot. We have started to experiment with some virtual reality headsets and wide angle cameras with the hope of driving the robot would get closer to a cockpit-like experience and the operator could navigate better in confined spaces if he/she could look around effortlessly. This approach also reduces the number of cameras required and the number of moving parts in the system as panoramic cameras do not need the pan-tilt mechanics.

4) Engineering improvements: Our unique robot chassis design allows us to change the wheelbase of the robot when required, however implies some structural challenges when driving on rough terrain. We have concluded that the concept of the 2016 robot works very well in the RoboCup Rescue environment, only some minor changes are required to make the structure more rugged. We have optimized the robot by replacing some aluminum parts with steel or stainless steel, have changed some steel parts with plastic where weight has significant importance. We have also optimized the robot-geometry so it has better cornering abilities and also can be more stable when required. The overall optimization resulted in a more compact, more agile, faster, stronger robot, that gives us the confidence in extreme missions like step-fields or stairs with debris.

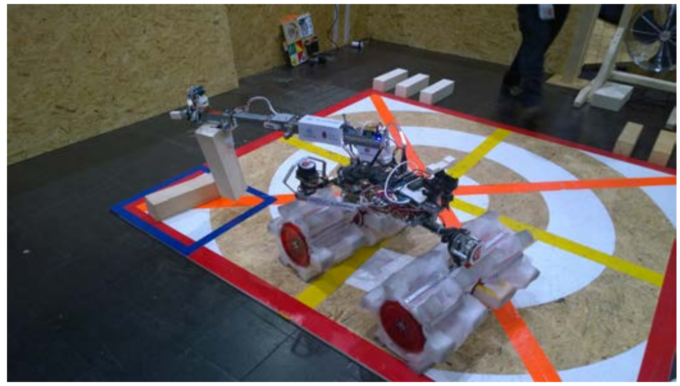

Fig. 2. Shoring task

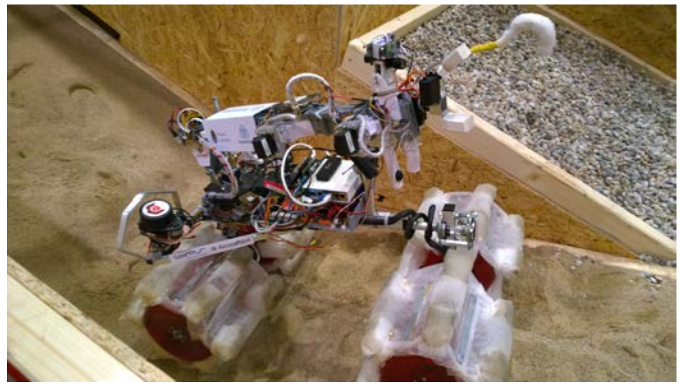

Fig. 3. R16 robot in the sandpit

Our 2017 robot fits the BOLDDAL: Small Robot robot-class, passing through a 60×60 cm square cut-out is possible now.

We have made efforts to make the operator station more compact and portable by one person but having multiple displays and directional antennas result in increased weight and volume.

In 2016 our robotic arm allowed us to score 10/10 in the vast majority of the several readiness tests at the beginning of each task, however the strength of the plastic arm had a lot of space for improvements. Also as the rules have changed and heavier wooden blocks are now the normal in the shoring tasks, we had to redesign the arm from scratch. The new arm is stronger in every aspect: can lift up 2.5 kgs of weight, has greater action radius and in retracted position the overall height of the robot is reduced as well. We have a hybrid approach to the arm control where the operator works in working planes not in polar coordinates, inverse kinematics also helps achieving precise movements. The arm is more dexterous so we believe in readiness tests finished quicker and also gaining more points

Fig. 4. Climbing a steep staircase with debris

in the dexterity challenges as well.

II. System Description

Our credo is that robots should serve a purpose. The RoboCup Rescue competition defines the scope we are focusing on, and with every development iteration we use the gathered information to optimize the current configuration. Of course there are some general key decisions we had to make ahead, which must remain the same throughout the development. The primary guidelines and decisions are:

- rigid body needs more weight thus forces the powertrain to scale up, making the robot expensive and failure prone (as tracks require a rigid body, we have opted for elastic chassis with all wheel drive and wheels optimized for the task)
- flexibility, torsion and deformation should be a design goal. The robot might lose some features while on mission, but must never stop moving. The planned sacrifice of replaceable parts could save the day in a competition or save a life in real disaster situations.
- minimizing the static surface under the robot (if only moving parts could ever touch the terrain, the only constraints of movement are the traction, center of gravity and motor power)
- dynamic wheel configuration (that way the robot would always have a degree of freedom to move, even when some wheels do not have traction)
- keep everything simple. We are working hard to never add parts without a good reason and we are trying to keep the system complexity at the minimum
- there is a defined powertrain cascade: a planned series of graceful degradation in the wheel-gear-motor-esc-battery line. When stuck it is acceptable to lose a part from the wheels, or even lose a whole wheel, but it is not acceptable for a motor controller to fail at the maximum rated power consumption of the motors. So each part in the chain is more durable than the previous item, and this way the robot could still operate with reduced features
- every part must be accessible and repairable/replaceable on site: we try to avoid using very specialized manufacturing techniques and choose the simpler solutions, which we can replicate in the competition area in reasonable time
- as search and rescue robots are considered “expendable” in a disaster scenario, a cheaper solution for the same problem would give the ability to use more robots at the same time. So we are trying to use cheap off-the-shelf parts and avoiding the dependency on expensive sensors.
- we are developing the hardware in an agile environment (adopted from software development), so we do not spend months on cad and planning. Instead we iterate every week and we are planning and building proof-of-concept robots in a continuous pipeline, in order to validate the ideas. Then we apply the lessons in the next design phase and we test the next generation accordingly. This method allows us to always have working prototypes and parts thus not risking the not-being-ready-to-move situation a waterfall based project often results in.

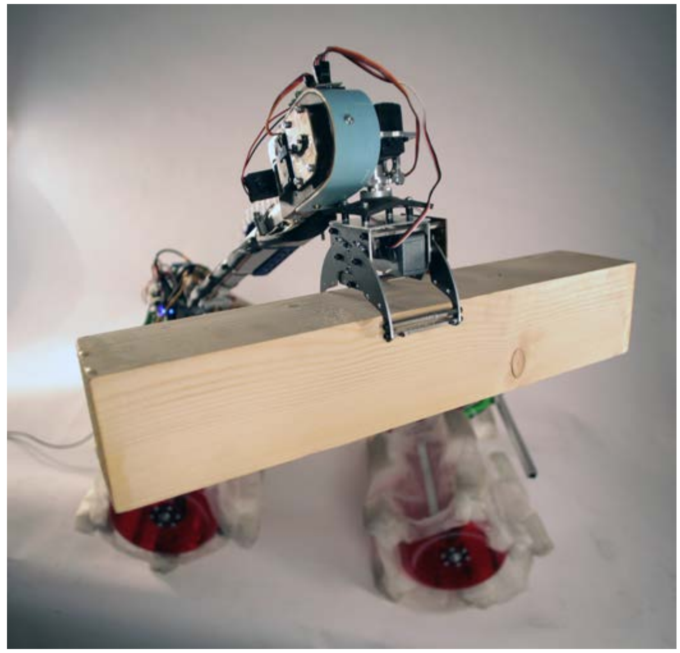

Fig. 5. The 2017 prototype robot holding a 2.5 kg woodblock

A. Hardware

The primary goal is to never get stuck. The all wheel drive construction with no other surface to touch the ground combined with the ability to change the wheelbase and move the center of mass gives the robot extreme maneuverability (Fig. 4). The second task is to learn how to efficiently drive/control this configuration to solve the tasks.

A *static wheel hub* design (Fig. 6) allows us to use larger wheels with double bearing and also to protect sensitive parts inside the rotating surface. With the ability of quick wheel replacement, the vehicle can always be adapted to specific terrains and challenges in minutes (Fig. 3).

For *motors and gears* we were using cheap off-the-shelf servos in the early prototypes, but after a few iterations we gave up as we calculated the speed and torque requirements to drive our large and exotic wheels in every situations. In

Fig. 6. Static wheel hub design

Fig. 7. Team Rescube in RoboCup Championship 2016 Leipzig

2015 and 2016 we found the Banebots 256:1 planetary gears and their motors to match our needs perfectly and they worked so well, we will continue using them in 2017 too.

For high performance *on-board computing* we have decided to use multiple Odroid XU4 single board computers as they have excellent computing power compared to their size. This versatile 8 core ARM-Linux computer runs a fully fledged Linux with ROS and has abilities to run the on-board processing required on missions.

For *low-level operations* and where real-time operation is a requirement we use TI's Tiva-C ARM-Cortex M4 microcontroller. This excellent MCU has a great variety of hardware peripherals and is responsible for collecting data from various low-level sensors and also to command the drive motors. The communication between the Tiva and the Odroid is handled by the Rosserial ROS-package.

We have decided to use Lithium-Polymer high discharge *batteries* for their superb capacity/price ratio. The total on-board capacity is 10 Ah providing the robot over 30 minutes of continuous operation in normal scenarios. We use several high-efficiency voltage converters to supply the adequate voltages/currents for each subsystem.

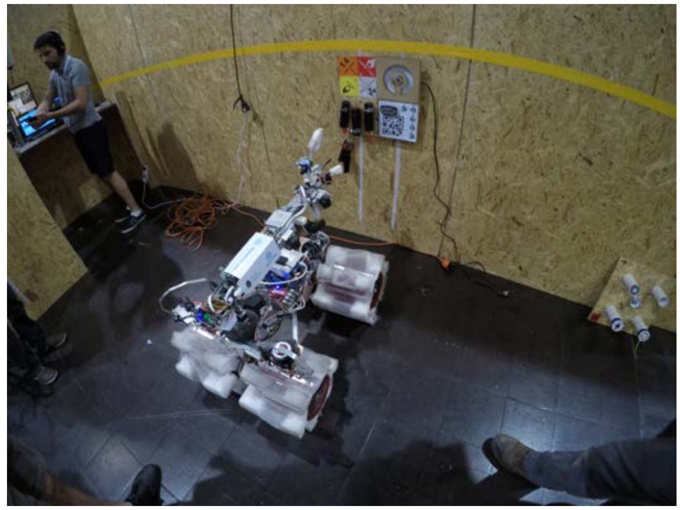

Fig. 8. RoboCup WorldCup 2016 readiness test

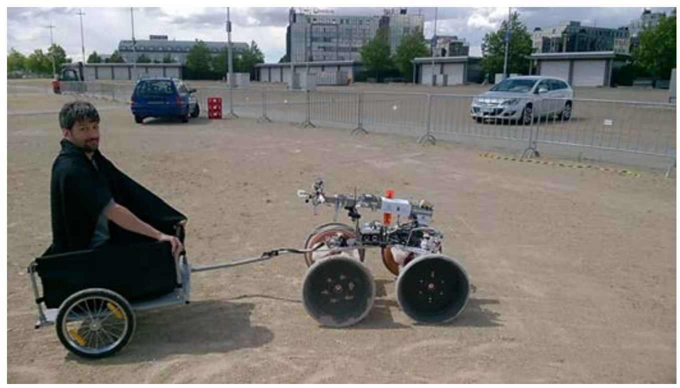

Fig. 9. RoboCup WorldCup 2016 Best Outdoor CarryBot Award

Fig. 10. The 2017 arm is much stronger and more versatile

The *robotic arm* works like a crane with a rotating base and an extendable telescopic arm that has strong actuated grippers to enable the robot to open doors, turn valves and execute shoring tasks (Fig. 10).

For *cameras in visible light range* we plan to use Sonys PS-Eye cameras which have a well deserved fame among developers for their excellent frame-rates and low-light sensitivity. Since 2016 we have upgraded our lidar to the newer RP-Lidar A2, which has higher sampling rate, greater range and precision, however the most important difference is the mechanical build quality compared to the previous version.

For outdoor missions our laser scanner data is fused together with the measurements of a u-blox M8T multi-GNSS receiver providing 10 Hz absolute positions. To achieve centimeter-level accuracy we plan to establish a short-baseline RTK system by operating a reference station in situ.

A structure.io *depth camera* gives us 3D perception abilities and it provides the information needed for detecting 3D structures, and movement. We will continue to use Flir's Lepton thermal camera sensor modules for their outstanding capabilities. We have been one of the first Lepton users in the Rescue League and Flir recently showcased our project on their community page (https://lepton.flir.com/community-showcase-items/robocup-rescue-league-robot).

For *inertial measurements* and pose sensing the Invensense's MPU-6050 6-DOF IMU coupled with a Honeywell HMC-5983 compass are still top-notch devices in their categories despite their age.

Of course the robot also has some extra sensors/devices like microphone arrays, speakers, a CO2 and an air pressure sensor, and numerous robot state encoders/potentiometers built in. We primarily use the Tiva-C microcontroller boards to communicate with these low-level digital (over I2C and SPI) or analog devices, or USB for the higher level protocols.

Please see Tables I, II as well as Table III in the Appendix.

B. Software

The general software framework in use is the open source Robot Operating System, ROS, version Indigo[1].

For simulation our intention is to use Gazebo 4 simulation platform enabling us to quickly and easily test new design concepts and software features[2].

For mapping we have decided to use the Hector SLAM module of ROS developed by Stefan Kohlbrecher and Johannes Meyer at TU Darmstadt[3].

For GNSS positioning we have good experiences with Tomoji Takasus RTKLIB, an open source GNSS positioning software[4].

For pose calculations using inertial sensors we use Sebastian Madgwicks open source implementation of Robert Mahonys DCM filter[5]. We have plans for developing an own-rolled EKF solution for sensor fusion using Matlab.

OpenCV is an excellent and proven tool for image processing and feature detection[6].

The PCL (Point Cloud Library) contains routines for the manipulation of sets of discrete points, and allows the user to detect surfaces and predefined objects well[7].

Please see Table IV in the Appendix.

[1] http://www.ros.org/
[2] http://gazebosim.org/
[3] http://wiki.ros.org/hector_slam
[4] http://www.rtklib.com
[5] http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms
[6] http://opencv.org/
[7] http://pointclouds.org

C. Communication

For communication between our hardware nodes not connected physically we rely solely on TCP/IP commu- nication over wireless 802.11ac compliant network devices running on 5 GHz channels. Our network setup allows for automatic transparent failover to 2.4 GHz and then to 4G/LTE broadband connectivity. We are using 120 degree directional sector antennae for the 5 GHz and 2.4 GHz channels, which allows us to lower the signal to noise ratio, and also this way we will not interfere with other teams. When placing ROS nodes on computers our goal is to keep networking load at the minimum, so every computer should process the information of their connected sensors and just send the deducted information to the other interested parties. We have also developed a solution of a low overhead mission logging concept that collects the data required for re-enacting the missions while keeping network load minimum.

D. Human-Robot Interface

The Human-Robot interface is recognized as a bottleneck so we had developed some optimization to help the operator focus on the tasks at hand. We have modified some ROS-modules like rqt_image_view to reduce the latency and over-processing of image streams so some processing like rotating and overlaying is actually happening at the display node, thus having very little overhead. Our experimental head-mounted display setup would give on-board panoramic view to the operator for driving and also for dexterity tasks.

III. Application

This section covers the practical aspects of our system.

A. Set-up and Break-Down

All our robots and operator station was designed with quick and easy deployment capabilities in mind. The typical set-up time for robots is under five minutes, the time requirement for the operator station is around ten minutes, so our complete system can achieve operational state in about 10 minutes (assuming the processes executed in parallel). We also have made efforts to let the robot and operator station work uninterrupted (even when replacing batteries), so no cold restart is required between missions at all.

The break-down process consists of steps with more or less the same time requirements. The overall break-down process should take no more than 5 minutes under normal circumstances.

B. Mission Strategy

Our intention is to develop a versatile robot being capable to compete successfully in all Robocup Rescue standard scenarios, but with the prioritized order of: mobility-manipulation-autonomy. We believe that autonomy is a great feature to decrease the mental load of the operator (and to enable a single person to operate multiple robots). However, at disaster recovery the human decision making could save lives, so we strive to create a hybrid semi-autonomous solution that allows human and computer minds collaborate in a well-balanced and effective way.

C. Experiments

For evaluating our robots' performance we have tried to replicate some of the key elements of the common mobility and perception tasks based on the NIST standard test methods[8].

As we follow the agile methods in hardware developments, experimental features always start with a discussion, a proof of concept and then multiple versions are tested in realistic environments. A perfect example is the wheels of which we have already made 10-15 variations and we are still working on continuous improvements.

D. Application in the Field

While working on the development of RoboCup Rescue robots, our team members continuously learn new skills and apply those either in their academic, industrial or personal life. We are dedicated to share these experiences with the following young generations of engineers, so we have an educational/motivational mission, to give them reason to learn programming, to solder leds to a microcontroller, or to learn the math required for every applied scientific career path.

IV. Conclusion

As a new RoboCup Team we do have a lot to learn, and we are still working hard to create a community in Hungary where robotics interested people can gather together. We believe that our efforts could bring industry, academia and elementary/high schools together, and create an accelerated path where applied science students could gain invaluable experience every day.

Appendix A
Team members and Their Contributions

- Zoltán Abonyi embedded computing, robot operator
- Mátyás Borvendég electrical engineering, manufacturing
- András Czikora mechanical engineering
- Dávid Dudás mechanical engineering, 3D printing
- Péter Gliga electrical engineering, manufacturing
- Péter Kopiás team leader, robot concept, software development
- Márton Krauter localization, inertial measurements, public relations
- Miklós Márton software development, embedded computing
- Zoltán-Csaba Márton 3D perception, object recognition
- Imre Petrovszki electrical engineering, software development

Appendix B
CAD Drawings

Please see Fig. 11 and 12, along with the earlier drawings.

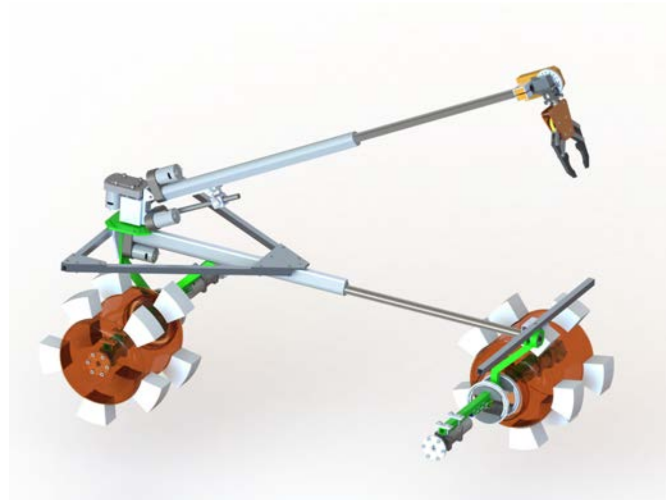

Fig. 11. Render view of the new robot (1)

Fig. 12. Render view of the new robot (2)

TABLE I
MANIPULATION SYSTEM

Attribute	Value
Name	R17
Locomotion	4 wheel drive
System Weight	20 kgs
Transportation size	100 x 100 x 50 cm
Typical operation size	100 x 59 x 45 cm
Unpack and assembly time	10 min
Startup time (off to full operation)	5 min
Power consumption (idle/typical/max)	50 / 240 / 4500 W
Battery endurance (idle/normal/heavy load)	4 / 1 / 0,5 h
Maximum speed (flat/outdoor/rubble pile)	5 / 5 / 3 km/h
Payload (typical/maximum)	3 kg / 20 kg (200kg towing capacity)
Arm: maximum operation height	160 cm
Arm: payload at full extend	2500 grams
Support: set of bat. chargers total weight	1 kg
Support: set of bat. chargers power	160 W
Support: Charge time batteries (80% / 100%)	0.5 / 1 h
Support: Additional set of batteries weight	1 kg
Any other interesting attribute	Dynamic wheelbase
Cost	3000 EUR

TABLE II
OPERATOR STATION

Attribute	Value
Name	Operator station
System Weight	20 kg
Weight including transportation case	20 kg
Transportation size	80 x 60 x 30 cm
Typical operation size	80 x 80 x 100 cm
Unpack and assembly time	10 min
Startup time (off to full operation)	5 min
Power consumption (idle/ typical/ max)	200W
Battery endurance (idle/ normal/ heavy load)	120 / 90 / 60 min
Any other interesting attribute	Custom made box, with fold-out external display and controller panels
Cost	1500 EUR

TABLE III
HARDWARE COMPONENTS LIST

Part	Brand & Model	Unit Price	Num.
Drive motors	Banebots RS540	8 EUR	4
Drive gears	Banebots P60	60 EUR	4
Motor drivers	Non-brand	30 EUR	4
Smart servos	Turnigy S518D	40 EUR	6
DC/DC	Non-brand	30 EUR	2
Batteries	Turnigy 5000mAh	50 EUR	2
Battery chargers	Turnigy 80W	50 EUR	2
Microcontroller unit	TI Tiva C	20 EUR	2
Computing unit	Hardkernel Odroid XU4	100 EUR	2
Wireless network	Mikrotik Routerboard	500 EUR	2
IMU	Drotek 6-DOF IMU	15 EUR	2
Compass	Drotek HMC-5983	10 EUR	2
GNSS	Drotek u-blox M8T XXL	90 EUR	2
USB camera	Sony PS Eye	30 EUR	4
Infrared camera	Flir Lepton	200 EUR	2
Lidar	RPLidar A2	500 EUR	1
CO2 sensor	Non-brand	20 EUR	1
Operator laptop	Lenovo T430	1000 EUR	1

TABLE IV
SOFTWARE LIST

Name	Version	License	Usage
Ubuntu	14.04	open	OS
ROS	Indigo	BSD	Framework
OpenCV [1]	2.4	BSD	Computer Vision, image-based recognition
PCL [2]	1.7	BSD	Scene segmentation and object recognition
Hector SLAM [3]	0.3.4	BSD	Localization and Mapping
RTKLIB	2.4.3	BSD	Localization

APPENDIX C
LISTS

A. Systems List

B. Hardware Components List

C. Software List

ACKNOWLEDGMENT

We would like to express our gratitude for their contributions towards the following individuals and organizations:

- Datastart Ltd.
- Former members: Bence Magyar, Árpád Pálóczy, Sam Pfeiffer
- ABC Center for Intelligent Robotics: Tivadar Garamvölgyi, dr. Péter Galambos
- Óbuda University: dr. László Kutor, Gusztáv V. Tényi, Tamás Sándor, dr. György Schuster, dr. József Tick, dr. András Molnár, dr. Zoltán Vámossy
- John von Neumann Computer Society: dr. Zoltán Istenes
- Attila Sipos: http://magyarokamarson.hu
- Technische Universität Darmstadt: Prof. Dr. Oskar von Stryk, Stefan Kohlbrecher
- Robocup Rescue League Organizing Team
- AccessPoint Ltd. http://accesspoint.hu
- ThyssenKrupp Presta http://thyssenkrupp.hu
- Business Power Systems http://bps.hu
- Drone Prix Hungary http://droneprix.hu
- Hungarian Robot Builders' Association http://hurba.hu
- Optoforce http://optoforce.com
- Weisz Müanyagipari Kft.
- RobotShop Robotshop.com

[8]http://www.nist.gov/el/isd/test-methods.cfm

REFERENCES

[1] G. Bradski, *"The OpenCV Library,"*, Dr. Dobb's Journal of Software Tools, 2000.

[2] R. B. Rusu and S. Cousins, *"3D is here: Point Cloud Library (PCL)"*, in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011

[3] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, *A flexible and scalable slam system with full 3d motion estimation"*, in Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). IEEE, November 2011.