RoboCup Rescue 2016 Team Description Paper CUAS RRR

Patrick Hofer, Stefan Quendler, Martin Sereinig, Wolfgang Werth

Info

Team Name:	CUAS RRR
Team Institution:	Carinthia University of Applied Sciences
Team Leader:	Martin Sereinig, MSc.
Team URL:	https://fh-kaernten.at/robocuprescuerobot

RoboCup Rescue 2016 TDP collection:
https://to-be-announced.org

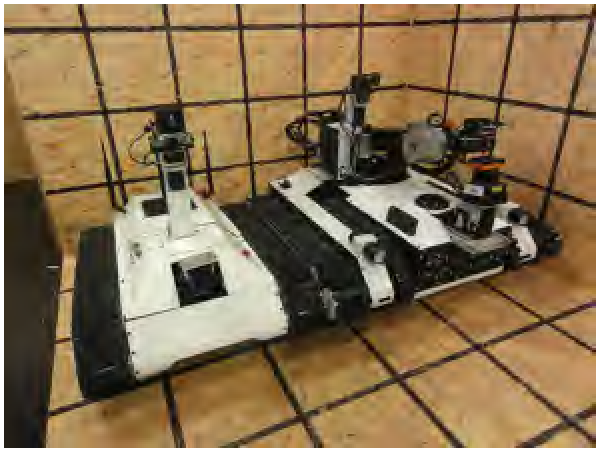
***Abstract*—This paper describes the robots as well as the approach regarding the RoboCup Rescue competition by the CUAS RRR - Team. The robots were designed and built for participating at the RoboCup Rescue competition by a team of researchers at the Carinthian University of Applied Sciences. The technical details about the communication structure, the control method and the human - robot interface are given. To make it easier for the user to navigate the robots, a graphical user interface including advanced mechanisms like a head up display was developed. The presentation of map generation and the sensor system for navigation, localization and victim identification is also a part of this paper. Furthermore the mechanical design and locomotion, with a special focus on the suspension system, chassis structure and flipper rotating mechanism is described.**

***Index Terms*—RoboCup Rescue Roboter, disaster scenario, rescue operation, USB data acquisition, suspension system, brushless motors, head up display, communication structure, chassis design, mechanical design, control systems, inverse kinematics, computed torque control.**

I. INTRODUCTION

DURING a disaster scenario, for example a collapsed building excited by an earthquake, it is often very dangerous for human emergency forces to reach a casualty. The risk for the rescue team is reduced and the survival chances of the victims can be increased with the aid of robots. Rescue robots can be operated in dangerous areas to find victims and supply essential medicine or food[1]. They are also capable of mapping the path to the victims if obstacles interfere the encroachment of the emergency forces. It is also possible to equip the robots with sensors to check for risks, dangerous gases, or to mount a heat sensor to find entombed or invisible victims. Of course there are also other scenarios where human emergency forces cannot perform the rescue operation without robots. An actual incident where robots were needed was the Fukushima Daiichi nuclear disaster. Because of the highly dangerous environment, robots were used to investigate the contaminated parts of the nuclear plant.

The RoboCup Rescue Senior competition is a simulated disaster scenario, in which the robots have to fulfil tasks similar to a real rescue operation. This paper presents the robots **R.U.D.I.** (**R**obot for **U**rban **D**isaster **I**tervention) and **R.A.U.D.I.** (**R**obot for **A**utonomous **U**rban **D**isaster **I**ntervention), developed at the **C**arinthian **U**niversity of **A**pplied **S**ciences (CUAS) as well as the approach of the CUAS RRR team and the contributions to the RoboCup Rescue Senior community. The discussion of the technical details is split in four chapters. General system description (II) contains the teams overall approach as well as communication structure, human-robot-interface, set-up and break-down procedures and experiments done. The three following sections describe R.U.D.I. (III), R.A.U.D.I. (IV) and the manipulation arm (V).



(a) R.A.U.D.I. and R.U.D.I. side by side.

(b) R.U.D.I. with extended manipulation arm and flippers.

Fig. 1: The two robots of the CUAS RRR team at the Magdeburg German Open 2014.

The two robots R.U.D.I. and R.A.U.D.I. which are actively in use for the RoboCup Rescue competition were developed and manufactured at the CUAS [2],[3],[4],[5],[6]. Controlling the robots is accomplished with the help of four subsystems, two windows based systems for R.U.D.I. and two Linux based systems for R.A.U.D.I.[7],[8],[9].

A. *Improvements over Previous Contributions*

The first participation of the team at the RoboCup was at the RoboCup German Open in year 2013[10]. There the team of the CUAS started with the remote controlled Robot R.U.D.I. and accomplished the third place. At the RoboCup German Open in 2014 the team participated with two robots. There the robot R.U.D.I. and the new developed autonomous robot R.A.U.D.I. where used in cooperation. The robot R.U.D.I. was equipped with a new manipulation arm. With this setup the team was able to achieve the third place again. In 2015 the researchers developed a new control strategy, including a new control algorithm[11] and an inverse kinematics algorithm[12] for the manipulation arm. The current work will focus on the collaboration between the two robots. This will improve the performance at the proposed participation at the RoboCup WorldCup in 2016.

II. GENERAL SYSTEM DESCRIPTION

The team of researchers at the CUAS developed two different robot platforms. The remote controlled robot R.U.D.I. was designed to drive to rough terrain and overcome higher obstacles, the second robot R.A.U.D.I. was designed to work autonomously. During a mission both robots should work simultaneous in the area to increase efficiency. The following sections describes the basic features for both developed robots, whereby the multi operating system approach, where Windows and Linux is used in combination to control the robots, is pointed out.

A. *Communication*

The communication between the operator station and the robots is done throughout the *WNDR3700* router from *Netgear*. It acts as a platform for remote controlled, partially autonomous and fully autonomous actions of the robots. It also supports true dual bands, offering simultaneous Wireless-N performance in both 2.4GHz and 5GHz bands. The 5GHz band is preferred basically, because of the low distribution and subsequently less utilized communication channels.

At software level the exchange of information is split into the *Winsock*-API and the ROS internal communication strategy. The data packets are transferred from the Operator station to the robots and vice versa, whereby both kind of sockets, TCP and UDP are used to establish connections in a convenient way. If it comes to videostreaming the transmission of faulty data packets or even a loss of data may be acceptable, so transmission via UDP was taken into consideration. On the other hand, if it comes to a transmission of control data, a secure and failure free connection should be guaranteed, which leads to the use of the more reliable TCP protocol. Connected to this increased transmission latencies and in average slower data throughput must be taken into account.

TABLE I: Communication data

Rescue Robot League		
CUAS RRR-Team (AUSTRIA)		
Frequency	Channel/Band	Power (mW)
2.4GHz - 802.11n	11	100
5GHz - 802.11n	48(TPC)	200

The communication between both robots was established by using ROS standards. Therefore ROS was installed on the Windows7 OS which is used on R.U.D.I. and on the robot remote system.

B. *Human-Robot Interface*

The interaction between operator and robot is achieved throughout the use of a multifunctional joystick. Further the joystick is coupled with the robot remote system (RRS) respectively a *HP Elitebook* laptop which is connected to the robots control system via WLAN. As part of the RRS, a graphical user interface provides all necessary informations of the robots current condition to the operator. These are e.g. live camera view for navigation but also basic status information like orientation of the robot in the plane.

1) Input Device: The input device is represented by a *Thrustmaster Hotas Warthog* joystick. On one hand it possesses multiple rotational and linear axis and on the other hand various buttons make the device a powerful functional user input platform. Interfacing with the joystick is done via USB and further on DirectX is used for a windows based application development.

2) Graphical User Interface: For both robots a graphical user interface (GUI) was developed where all useful information for the operator is shown. The GUI for R.U.D.I. is a C# based application running on a Windows platform, which is shown in figure 2. The GUI for R.A.U.D.I., a C++ based application running on a Linux platform, can be seen in figure 3. Both GUIs contain the following informations:

- List of basic sensordata (e.g. power supply status, orientation of the robot in the plane, status of motor controller boards, etc.)
- Live camera stream of the robot for navigation
- 3D-View of the robot inside the arena
- Current status/overview of acquired 2D-mapping
- Menu structure for triggering shortcuts and administrative tasks

It is difficult to filter and display the data, which is received from the robot, in a convenient way. Therefore, in addition to the foregrounded interface, where important information is shown only, a back-end with all collected data is implemented. So the operator is able to concentrate on the most important information while controlling the robot in a conventional way. On the other hand if failures occur it is possible to investigate on the saved data where detailed status information of the robot can be extracted.

For the remote controlled robot a head up display (HUD) as shown in figure 4 makes important data available at first sight. Based on concepts which are widely used in aircrafts, like jet fighters, the following content has been implemented:

Fig. 2: Overview of the current graphical interface used for R.U.D.I.

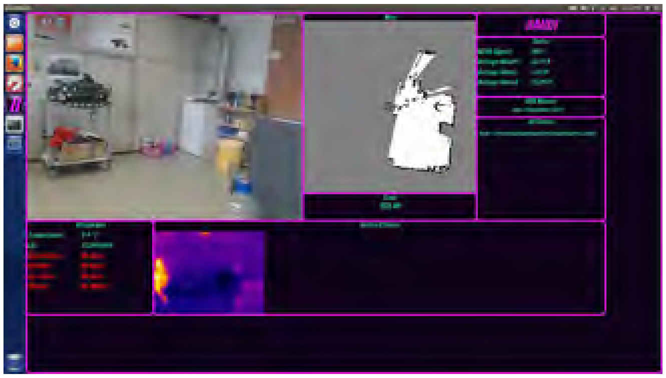


Fig. 3: Overview of the current graphical interface used for R.A.U.D.I.

- Horizon (A): The horizon basically represents the current offset nick-angle from the zero-level. This offset is updated in real time and scaled by horizontal lines which are drawn over constant intervals. Throughout this, the horizon is able to provide the operator with important orientation information while climbing up or moving down stairs or getting over some obstacles. Providing information about roll- and pitch-angle are taken into consideration.
- Flipper-position (B): Another important type of information, which has to be reachable as fast as possible, is the position of the flipper arms. It is represented by elliptic trajectories, whereby a vector points to the current position.

It is noticeable that the elements which are used inside of the HUD, may seem simple at first sight. On the other hand especially this circumstance contains a lot of capabilities, because the designer is forced to accommodate information in a simple way, concurrently concentrating on most important elements only. For example, by the use of just two more mechanisms (besides of the form), color and thickness may extend the prospects of information-representation by a multiple.

C. Set-up and Break-Down

The operator station consist out of a steel construction equipped with wheels for a easy transport. This construction is adjustable for different operators and different environmental conditions. Furthermore it offers enough space for the remote robot control unit, the router and the grid independent power supply. The remote robot control unit of the rescue robot consists of two HP Elite notebook where the remote applications including the graphic user interfaces are running. Beside this mode of operation an additional input device, the joystick, is needed. The entire remote control unit is portable and can be set up by a single person.

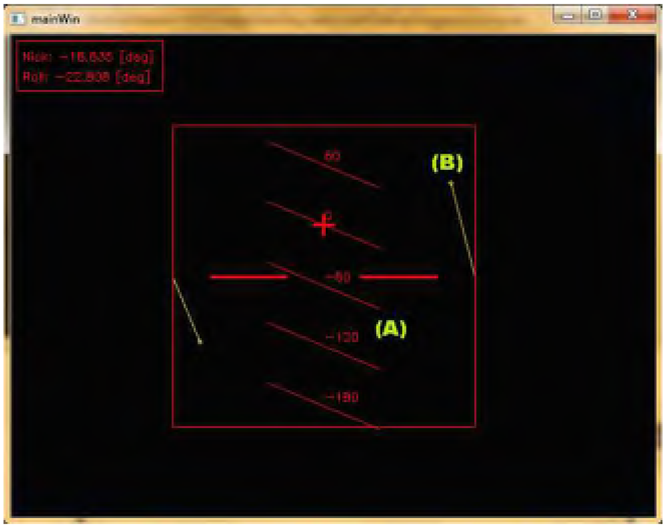


Fig. 4: Overview of the HUD.

The robots itself can be transported in a box armed with wheels. The start-up sequence is simple and the robots are ready to use in about ten minutes after it is placed at the operation area. The connection between robots and remote systems is established automatically.

D. Experiments

At the CUAS a special test laboratory was implemented where different testing areas can be used according to the RoboCup Rescue rules. Furthermore tests under different environmental conditions (indoor and outdoor) were made. Therefor the robots were tested on a scrap yard, under urban conditions and in the forest as shown in figure 5 and figure 6.

III. SYSTEM DESCRIPTION: R.U.D.I.

R.U.D.I. is a tracked mobile robot, equipped with a mounted manipulation arm as given in figure 1. The representation of the robot in figure 8 helps to discuss the robots subsystems and their position within the chassis. The main drive of the robot consists of two $200W$ brushless DC electric motors which powers a track (A) with rubber pads. This track is suspended by a spring-damping element to increase overall mobility and drive performance. The battery storage (B) contains lithium-polymer batteries. Two cell packs, ten cells each, provide the energy for locomotion and manipulation of the arm drives, while two separate packs (6S-packs) power the computational unit. With the help of the gearbox (C) two flipper arms (D) can be rotated independently by 360°. Flipper arms are used

Fig. 5: R.U.D.I. tested on a scrap yard

Fig. 6: R.A.U.D.I. tested under urban conditions

to overcome obstacles like steps and can be driven standalone from the main drive. The main computing element is an *Intel i5* processor mounted on a Mini-ITX mainboard located below the louver (E). On-Off switches and the kill switch are mounted on a operating panel. The on-off switches are only accessible when the cover panel (F) is lifted. A modifiable cover (G) allows the clean mounting of additional systems. Although R.U.D.I. is designed especially for the RoboCup Rescue Senior competition, where a manipulation arm is essential, the robot can be used as a carrier platform for diverse systems.

A. *Mechanical Design*

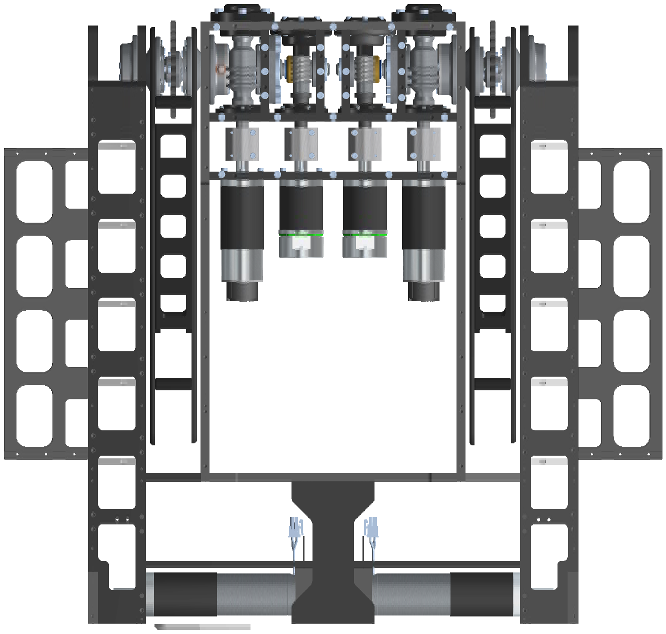


Fig. 7: Top view of R.U.D.I.s partly-assembled frame and gearbox.

R.U.D.I.s chassis is composed of a frame, manufactured using engineering plastics, and steel sheet metal parts. The *TECAMID6 GF30* plastic is reinforced with 30% glass fibre. The parts where designed using the software *Solid Edge* and produced with CNC machines. Therefore they represent a convenient and precisely manufactured foundation for more complex systems like the flipper gearbox and main drive suspension system. The frame construction together with six electric drives and the flipper gearbox is given in figure 7.

Steel sheet metal parts can be manufactured using laser-cutting technology and formed using sheet metal bending machines. Some work-pieces are welded together to augment rigidity of the overall design. They are mounted to the plastic frame using screws, in order to increase stiffness of the robots chassis. Hence the sheet metal parts can be disassembled from the frame, access for maintenance can be reestablished quickly. Another core feature of R.U.D.I. is the suspension system, composed of six ground roller groups per main drive, one of these groups is explicitly seen in figure 9. Two ground rollers (A) are connected with an axis (E) and hold in place with circlips (D). Roller bearings are fixated in the mounting plate (I) with bearing cups (B), axial fixation of the shaft (H) is also realised with circlips (C). A mechanical stop (G) limits the rotational freedom of the connection rod (F). Using another connection rod (N) the mounting of the spring-damping element (M) is realised. A slide bearing (K) holds the

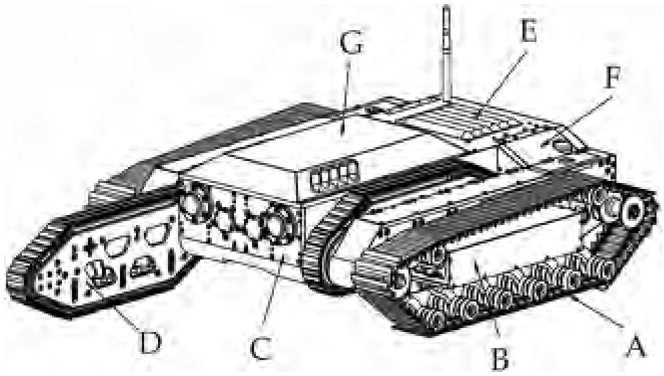


Fig. 8: System Description of R.U.D.I.

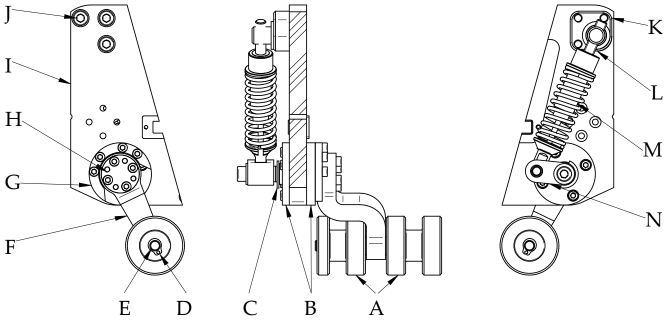


Fig. 9: Sectional view of a single suspension element.

upper mounting point (L) of the element. R.U.D.I.s suspension system allows a smother ride compared with purchasable mobile robot systems of similar size. The spring-damping elements are inside of the robot, isolated from dirt and moisture by the mounting plate (I).

Surface treatment of the steel sheet metal parts, the aluminium parts and the steel flipper gearbox shafts is realised with powder-coating, anodic treatment and black-oxide finishing [3], [5].

B. Software Design

Controlling R.U.D.I. is accomplished with the help of two large subsystems. The robot remote system (RRS) and the robot control system (RCS). Noticeable about RCS is the operating system, here Windows 7 Professional is used. Furthermore the RCS interfaces with USB based hardware modules from the company Phidgets [13], [7].

1) Robot Remote System: The RRS is used on the operating stations laptop to compute input commands from the operator and display informations provided for the operator. Control commands are transferred cyclic to R.U.D.I. via WLAN. Error message and safety related mechanisms are also included.

2) Robot Control System: The RCS is a soft realtime control system for the mobile robot. A Mini-ITX mainboard with an Intel i5 processor, SSD drive and 16GB Ram provide enough computational power to handle the high demand of RoboCup Rescue tasks. A optimized *Windows 7* with 64-bit architecture is the basic operating system, executing the C++ coded Win32- console application RCS [14].

IV. SYSTEM DESCRIPTION: R.A.U.D.I.

R.A.U.D.I. is the second mobile robot platform developed at the CUAS, again seen in figure 1. It presents an evolutionary step which advances the teams capabilities in search and rescue operations. R.A.U.D.I. was designed as autonomous robot, therefore it's main application environment within the competition is the yellow arena. Track suspension technology used for R.U.D.I. was developed further to improve ride quality in difficult terrain for R.A.U.D.I.. The tracks are driven by 200W brushless DC motors. Figure 10 shows the robot platform without metal sheet top covers and with open battery storage cover. A *HOKUYO UTM-30LX* laser scanner is mounted on a levelling system and fixated to the battery storage cover. In ready-for-operation configuration this clap is fixated with thumb screws to the chassis. The main computing unit is a Mini-ITX mainboard with *Intel i7* processor. The battery pack consists of four separate lithium-polymer cell packs providing energy supply for computing unit and drive.

Fig. 10: Rendering of R.A.U.D.I.s CAD model with opened battery cover and removed sheet metal covers.

A. Mechanical Design

R.A.U.D.I.s chassis structure is mainly composed of steel sheet metal parts which are screwed, welded and riveted together. Engineering plastic parts are only used as base plates to provide precise mounting points for the suspension- and drive-train-system, but are no primary load-bearing parts within the frame structure. The self-restriction of only using steel sheet metal parts for the frame offers faster production cycles and therefore lower manufacturing costs compared with the compound chassis frame of R.U.D.I..

R.A.U.D.I.s suspension system is a refined version of R.U.D.I.s.. Figure 11 shows six spring-damping element with ground rollers. Compared with R.U.D.I.s suspension system, large diameter ground rollers are equipped, which implies a low-pass filter effect reducing the oscillations introduced by the roller chain track. Spring-damping elements are outside of the robots chassis simplifying the mechanical structure of the suspension system (compare with figure 9).

An extension arm provides an elevated mounting point for important victim detection sensors like thermography- and video camera. The pan/tilt system is realized with Dynamixel Servos[15].

B. Software Design

The complete software of the autonomous robot is based on ROS hydro for Ubuntu 12.04. As main control unit, a finite state machine, written in C++, is implemented. The flow-chart of the state machine is figured in 12. Based on the current sensor readings, the state machine decides on the next task of

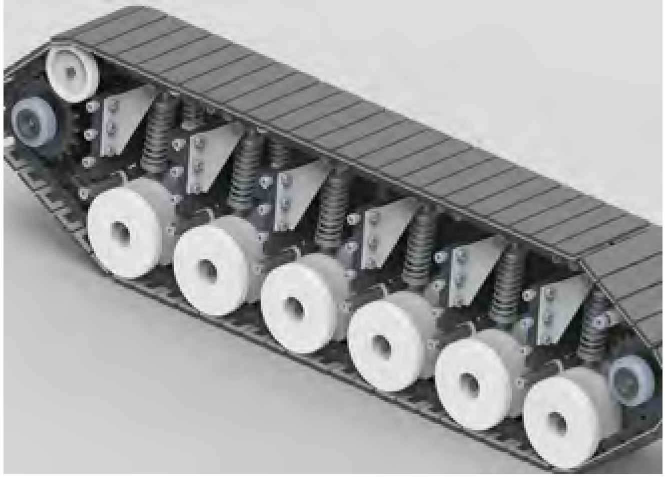


Fig. 11: Rendered side-view of R.A.U.D.I.s suspension system with dissembled side mounting plate and cover sheet.

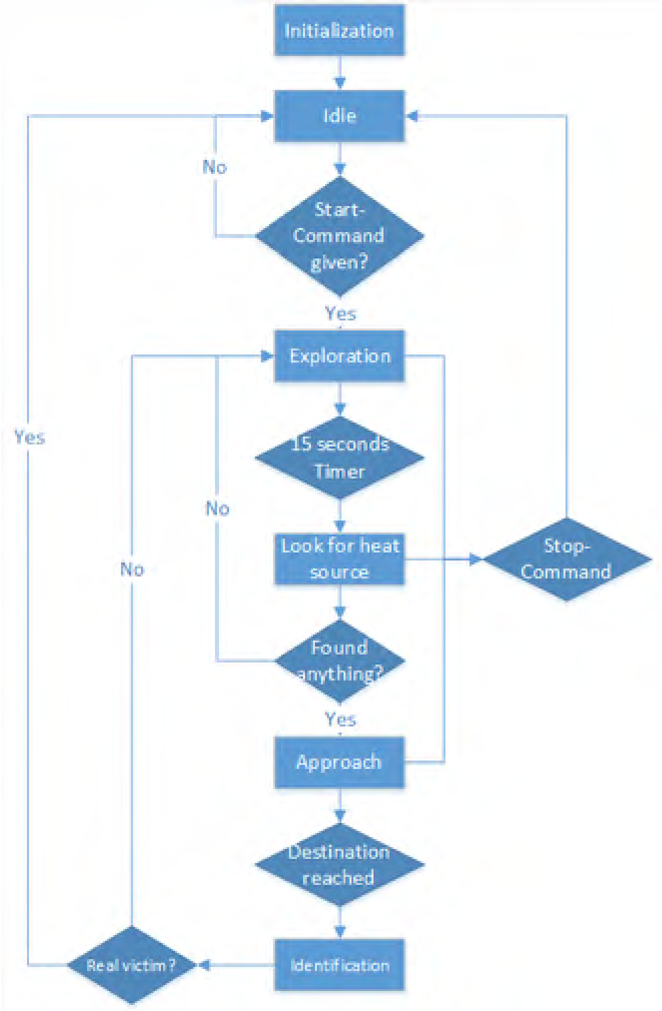


Fig. 12: Flow-chart of the finite state machine used for R.A.U.D.I..

the robot. In the first state, the robot explores the unknown area surrounding it. To localize itself in the environment and for the mapping purpose, the Hector Slam algorithm has been implemented. As exploration algorithm, the Hector Navigation packages are used[16]. After a set amount of time, the robot stops with the exploration and changes to the next state, which is used to scan the surrounding area for possible victims. If a heat source is detected via the IR camera, the position of this source is computed and the robot changes to the approach state. If nothing is detected during the scan, the robot continues with the exploration. Once the previously detected heat source is reached, the robot starts with a further investigation of the supposed victim. Therefore, the different sensor readings are checked for alternative victim criteria, like QR charts or movement. If at least one additional criteria to the temperature is detected by the robot, a signal is sent to the operator station, stating that a victim has been found. Once the operator acknowledges the found victim, the robot starts again with the exploration process[9],[17].

V. SYSTEM DESCRIPTION: MANIPULATION ARM

Many different tasks in a catastrophic scenario require the manipulation or inspection of places which are difficult to reach for a mobile robot. To ensure the fulfilment of those tasks a universal mountable manipulation arm was developed. Due to its flexibility different mobile robots can be equipped with this robot manipulation arm with six degrees of freedom. The mechanical structure was keep light and small in order to achieve a low profile of the robot system. This ensures that the whole mobile platform keeps its ability to operate all-terrain and also in low tunnels[18].

A. *Mechnical Design*

The arm itself consists of three main assembly groups as it can be seen in figure 13 . The base assembly group is visualized by green color, the planar part of the arm by yellow and the end effector by blue. The base ① is screwed to the chassis of the robot. To provide a rotary movement of the manipulation arm the Brushless DC motor ③ moves the swivel platform ②. Three Harmonic Drive motors ⑤ provide the motion for the planar links of the manipulator arm ④ . The end effector ⑥ is only drawn schematically.

To ensure small dimensions in folded position on the robot and a large working area, a redundant kinematic chain structure was chosen. The links were build of carbon fibre plastic to guarantee a lightweight construction. Due to the chosen rectangular profile the stability of the robot arm is sufficient.

B. *Software Design*

The software needed for manipulation of the proposed arm can be divided into three different sections. These are the main control algorithm design and the inverse kinematic algorithm. The third part is related to the implementation of the developed algorithms on the robot. Figure 14 shows the basic structure of the developed overall control strategy of the robot manipulation arm. The developed algorithms where simulated and tested using Matlab/Simulink.

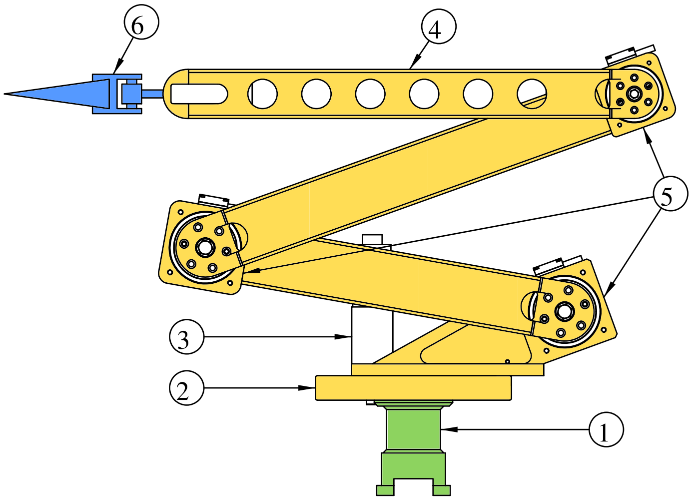


Fig. 13: Schematic sketch of the manipulation arm

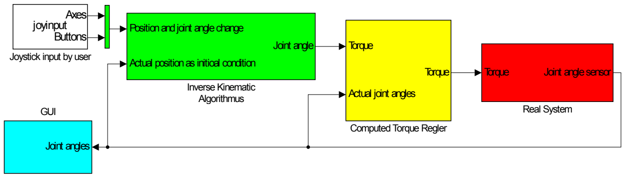


Fig. 14: Flow chart of the control strategy

1) Inverse Kinematics: To ensure a fast and easy usable movement of the robot arm in space a inverse kinematics algorithm was developed[12]. The main challenge of inverse kinematics is to calculate the joint variables θ with a given desired position and orientation of the end-effector.

There are three reasonable ways to describe the desired position and orientation with the help of a joystick input:

- The input is a point in the three - dimensional space, relative to a coordinate system fixed on the robot base. This method is often chosen for industrial manipulators. The fixed global coordinate system allows the operator to position the end effector in exact locations with certain orientation. However, there are problems when using this method for a mobile robot. Firstly, when operating the arm in a rescue scenario the end-effector goal position is not known beforehand, but has to be converged slowly by hand to avoid obstacles and keep the victims safe. Secondly, the operators video steam is provided by a camera mounted on the end effector of the manipulation arm. Therefore the operator does not have a visual information about the position in the global coordinate system.
- The input is a relative movement of a point in the three - dimensional space, relative to a coordinate system fixed on the robot. Hence the input is a speed the usage of the Joystick is intuitive. However, there is a problem when the orientation of the end-effector is changed. The point of view no longer aligns with the axis mapping on the joystick and therefore the operator is unnecessary challenged.
- The input is a relative movement of a point in the three - dimensional space, relative to a coordinate system fixed on the end effector of the robot. The point of view aligns with the joystick axis. If the orientation of the end-effector is changed, the view axis of the camera still aligns with end effector and therefore the operator is able to control the manipulation arm in a intuitive way.

Due to the redundant structure of the kinematic chain a closed form solution is not applicable to solve the inverse kinematic problem. Therefore a different approach using the combination of optimization algorithms was used. Here iterative optimization methods like Newton-Raphson, Broyden-Fletcher-Goldfarb-Shanno and the Cyclic Coordinate Descent algorithm by L.T.Wang and C.Chen were used [12]. The output of the described algorithm, which was designed for solving inverse kinematic problems, is used as input for the over all control structure.

2) Control Algorithm: To ensure the suitable motion in space a cascaded control structure is used. The electric motor of each joint is independently torque controlled by the used motor controller modules[11]. Furthermore a Computed Torque Controller (CTC), shown in figure 15, is used to provide the reference torque for motor controller modules. This nonlinear control strategy enhance the robot manipulator arm to have a good path following behaviour. In a CTC, the inertia M , the centrifugal C and the gravitation G depending matrices have to be calculated. Therefore the physical properties of the robot manipulation arm has to be known very well. The inner loop of the CTC is used to fully linearise the system, the outer loop includes a control law which defines a needed behaviour of the overall system [11].

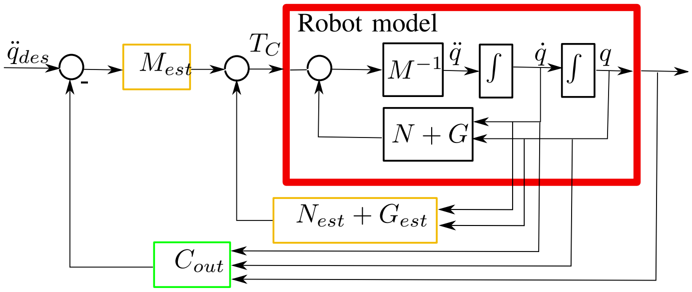


Fig. 15: Schematic drawing of the Computed Torque Control loop.

3) Software Implementation: The mentioned algorithms were implemented as C-functions generated by the Matlab/Simulink -Code generator. The developed RCS on R.U.D.I. provides the possibility to include those generated functions. Furthermore ROS nodes were implemented to insure compatibility with mobile robot platforms using ROS.

APPENDIX A
TEAM MEMBERS AND THEIR CONTRIBUTIONS

The RoboCup Rescue Robot R.U.D.I. was designed and developed by the first team between summer 2011 and summer 2012. The team in 2013 was responsible for the building process and the first test drives of the robot. Furthermore a manipulation arm was developed by a subgroup of the

TABLE II: Robot for Urban Disaster Intervention

Attribute	Value
Name	R.U.D.I.
Locomotion	tracked
System Weight	55kg
Weight including transportation case	60kg
Transportation size	0.6 x 0.62 x 0.5 m
Typical operation size	0.6 x 0.62 x 0.5 m
Unpack and assembly time	10 min
Startup time (off to full operation)	10 min
Power consumption (idle/ typical/ max)	120 / 300 / 540 W
Battery endurance (idle/ normal/ heavy load)	60 / 40 / 25 min
Maximum speed	1 m/s
Payload	15 kg
Any other interesting attribute	two additional drives (flippers)
Cost	25000 Euro

TABLE III: Robot for Autonomous Urban Disaster Intervention

Attribute	Value
Name	R.A.U.D.I.
Locomotion	tracked
System Weight	25kg
Weight including transportation case	30kg
Transportation size	0.6 x 0.4 x 0.45 m
Typical operation size	0.6 x 0.4 x 0.45 m
Unpack and assembly time	10 min
Startup time (off to full operation)	10 min
Power consumption (idle/ typical/ max)	120/ 300 / 450 W
Battery endurance (idle/ normal/ heavy load)	60 / 40 / 30 min
Maximum speed	1 m/s
Payload	1kg
Any other interesting attribute	3 DOF sensor head
Cost	10000 Euro

team members in spring 2013. In 2014 the new control strategy, including the inverse kinematics and overall controller algorithm, were developed. From 2015 up to now all robots were tested and optimised by the core team.

A. Core Team

- Wolfgang Werth, Supervisor, since 2011
- Stefan Quendler, Mechanical design and control systems, since 2011
- Martin Sereinig, Actuator and control systems, since 2011
- Patrick Hofer, Software and electronics design, since 2012

APPENDIX B
LISTS

A. Systems List

This section shows the properties of the main systems developed by the researchers of the CUAS. Table II and table III shows the main features of the two robots. In table IV the parameters of the robot manipulation arm were shown and in table V main properties of the operator station is given.

B. Hardware Components List

The table VI shows the main hardware components of the developed systems including price information.

TABLE IV: Robot manipulation arm

Attribute	Value
Name	Cuas Robot Arm
Locomotion	Joint actuators
System Weight	12kg
Weight including transportation case	15kg
Transportation size	0.7 x 0.2 x 0.2 m
Unpack and assembly time	10 min
Startup time (off to full operation)	10 min
Power consumption (idle/ typical/ max)	20 / 100 / 300 W
Payload (typical, maximum)	0.2/ 1 kg
Arm: maximum operation height	150 cm
Cost	9000 Euro

TABLE V: Operator Station

Attribute	Value
System Weight	20kg
Transportation size	1 x1 x 0.5 m
Typical operation size	1.5 x 1.5 x 1.5m
Unpack and assembly time	10 min
Cost	300 Euro

TABLE VI: Hardware Components List

Part	Brand & Model	Unit Price	Num.
Main drive motors	EC-4pole Motor 30 BL 200W	600 Euro	4
Flipper rotation motors	EC-max 40 BL 70W	425 Euro	2
Flipper drive motors	EC-max 40 Flat 70W	300 Euro	2
Motor robot arm	FHA 17 C	2500 Euro	1
Motor robot arm	FHA 14 C	1700 Euro	2
Motor robot arm	FHA 11 C	1600 Euro	1
Gears	Planetary Gearhead GP 42C	200	8
Encoder	Encoder HEDS 5540	80	6
Motor drivers	ESCON 70/10	250 Euro	4
Motor drivers flipper	DEC Module 50/5	60 Euro	4
Motor driver robot arm	ESCON Module 50/5	110 Euro	6
Batteries	Lippo Akku	100	10
Micro controller	Arduino Uno	35 Euro	1
DAQ Modules	1018 PhidgetInterfaceKit 8/8/8	70 Euro	1
Computing Unit	Mini ITX PC	500 Euro	2
IMU	Phidgets IMU	120 Euro	2
Cameras	FL3-U3-88S2C-C	1250 Euro	2
Infrared Camera	Optris PI 160	2500 Euro	1
Temperatur Sensor	1045 PhidgetTemperatureSensor IR	60 Euro	1
Laser Rangefinder	UTM-30LX	4500 Euro	1
Operator Laptop 1	hp elitebook 8760 w	2000 Euro	1
Operator Laptop 2	hp probook 650 g1	1500 Euro	1

TABLE VII: Software List

Name	Version	License	Usage
Windows	7	Professional	Operating System
Matlab	2014b	Education	Development and simulation
Ubuntu	12.04	open	Operating System
ROS	hydro	BSD	
ZBar	0.1	LGPL	QR detection
Optris driver	1.0	BSD	
Hokuyo Node	1.7.8	LGPL	
OpenCV	2.3.1-7	BSD	Victim detection
Hector SLAM [16]	0.3.3	BSD	2D SLAM
Hector Navigation	248	BSD	Autonomy algorithm
Siemens Solid Edge	ST7	Academic	CAD Software

C. Software List

In table VII the main software tools and programs can be found.

REFERENCES

[1] R. Sheh, A. Jacoff, A.-M. Virts, T. Kimura, J. Pellenz, S. Schwertfeger, and J. Suthakorn, "Advancing the state of urban search and rescue robotics through the robocuprescue robot league competition," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, K. Yoshida and S. Tadokoro, Eds., vol. 92. Springer Berlin Heidelberg, 2014, pp. 127–142.

[2] S. Quendler, "Konzept eines RoboCup Rescue Roboters," Bachelorarbeit, Carinthia University of Applied Sciences, 2011.

[3] ——, "Entwicklung des Fahrwerks für einen RoboCup Rescue Roboter," Bachelorarbeit, Carinthia University of Applied Sciences, 2012.

[4] M. Sereinig, "Aktorwahl, Modellierung und Regelung des Fahrverhaltens eines Roboters zum Einsatz im RoboCup Rescue Senior Wettbewerb," Bachelorarbeit, Carinthia University of Applied Sciences, 2012.

[5] S. Koch, "Entwicklung, Analyse & Realisierunge eines Hilfsantriebes für einen mobilen Such- & Rettungsroboter," Bachelorarbeit, Carinthia University of Applied Sciences, 2012.

[6] D. Lingitz, "Modellierung und Regelung der Aktorik eines RoboCup Rescue Roboters," Bachelorarbeit, Carinthia University of Applied Sciences, 2012.

[7] A. Isop, "Entwicklung und Implementation der zentralen Steuerungs-Applikation für einen "RoboCup Rescue" Roboter," Bachelorarbeit, Carinthia University of Applied Sciences, 2012.

[8] M. Ringswirth, "Entwicklung einer grafischen Benutzeroberfläche für einen Rescue-Roboter," Bachelorarbeit, Carinthia University of Applied Sciences, 2012.

[9] P. Hofer, "Analyse des Fahrverhaltens eines autonomen RoboCup Rescue Roboters," Bachelorarbeit, Carinthia University of Applied Sciences, 2014.

[10] A. Isop, S. Quendler, M. Sereinig, and W. Werth, "Team Desciption Paper CUAS RRR 2013," 2013.

[11] M. Sereinig, "Mathematic model and control concept of an articulated robot," Masterthesis, Carinthia University of Applied Sciences, 2015.

[12] S. Quendler, "Methods for calculating the inverse kinematics of a 6-dof rob ot manipulation arm," Masterthesis, Carinthia University of Applied Sciences, 2015.

[13] Phidgets, "Phidgets, products for usb sensing and control," http://www.phidgets.com/, 2013, [Online; accessed 06-Jannuar-2013].

[14] A. Isop, "Zentrales Steuerungssystem eines Rettungsroboters." Embedded Software Engineering Kongress, Sindelfingen, 2013.

[15] A. Brumnik, "Regelung von Dynamixel AX-18A Servos bei deren Verwendung als Schwenk- und Nick-Aktor des Kamerasystems eines RoboCup Rescue Roboters," Bachelorarbeit, Carinthia University of Applied Sciences, 2014.

[16] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, Nov 2011, pp. 155–160.

[17] A. Brumnik, "Implementierung einer OpenCV basierten thermographischen Opfererkennung für einen RoboCup Rescue Robot," Bachelorarbeit, Carinthia University of Applied Sciences, 2014.

[18] C. Unterweger, "Entwicklung eines Roboterarms mit Greifer für den Rettungsroboter R.U.D.I." ser. sterreichischer Tag der Mechatronik. Mechatronik Kongress 2013, 2013.