Journal of Network and Innovative Computing.
ISSN 2160-2174 Volume 4 (2016) pp. 011–019

RUDA - Robot for Search for Human Beings in Debrits and Avalanches

Luža R., Drahanský M., Zbořil F.

Department of Intelligent Systems - Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
iluza@fit.vutbr.cz, drahansky@fit.vutbr.cz, zboril@fit.vutbr.cz

Abstract: **The paper concerns about prototype of robot aimed on detection and rescue of people in debris and avalanches. First the paper describes robot modular architectue - the chasis and optional extension modules. Later the paper concerns about control system of the robot and its particular subsystems used for high level control and decision making.**
Keywords: robot, rescue of people, bioradar, debris, avalanche

I. Introduction

In todays world robots are still more and more common part of human lifes. We use robots for manufacturing, cleaning, in military operations and many other fields of human activity. Advantage of robots is that they can work with high precision and they do not need to rest. Another advantage is that in case robot gets damaged or destroyed there is no life lost - only material. Protection of health and lifes is probably the main reason to use robots in rescue operations. Robots can operate in dangerous areas that are too risky to be entered by humans. Robots can easily cary additional sensors and tools that are too large and heavy for human or that require precise positioning. Disadvantage of robots is that they have limited motion capabilities - they usually use track or wheel propulsion that has many limitations. Still usage of robots brings many benefits for rescue teams.

Using robots for dangerous missions is not a new idea. There are several teams in the world that develop robots for this kind of missions - for example robotic system called Orpheus [1] that is already used by chemical division of Czech Army during their missions, tEODor Robot developed by Telerob company [2] used by pyrotechnists or MOIRA2 robot [3] - the successor of inspection robot MOIRA. This paper describes project of universal robot aimed on rescue missions called RUDA. The RUDA is more similar to Orpheus or MOIRA robots - it is aimed on sensors more than effectors especially compared to tEODor robot that is equipped with a set of tools usable for removing layers of material and dismantling bombs. The main difference between RUDA, MOIRA and Orpheus is in used sensors. Orpheus is aimed on chemical pollution, MOIRA on remote inspection. RUDA has a set of sensors aimed on searching for people. Combination of sensors that RUDA carry is probably unique in world of rescue robots.

There is and ongouing research in rescue robotics topic. The scientists concentrate not only to the robots itself but also on improvement of algorithms for control of the robots. Important areas are localization, trajectory planning, mapping and also interfacing between robot and other members of a rescue team. Of course these topics overlap. Usually the recent solutions of robot navigation use localization in existing map ([4], [5]) or they create a map on-the-fly using SLAM approach [6]. Mapping and localization is important for almost any non-trivial task. For some missions it is importatn to explore entire area and cover it with robot trajectory - usually in search missions. Coverage of partialy observed area using SLAM to correct apriori knowledne of the environment is described in [7].

II. RUDA - Robot for Search of Human Beings in Debris and Avalanches

The robot called with acronym RUDA is being developed at Brno University of Technology at Faculty of Information Technology. The development team compounds mostly of students and their supervisors. The leader of the team is doc. Ing. Martin Drahanský PhD.

RUDA is a robot aimed on rescue of people in debris, avalanches and generally in dangerous situations. Its capabilities are limited compared to human but it can operate in environments that can not be entered by human. Example of such environments is avalanche field where is hight probability of avalanche landslide or environmend contamined by dangerous chemicals.

The robot is primarily aimed on detection of people in dangerous environments. Secondary it can help those people to get out from imminent danger. The robot can move around the operation area on tracks. The tracks are driven by two independent electro-motors powered from internal battery. Normally the robot is remote controlled. It communicates with operator via cable or via wireless data transfer. With cable connected the robot is also connected to source of power so it has theoretically unlimited operation time.

To provide good overview of surrounding situation the robot is equipped with two fixed cameras - front and rear. Moreover it has PTZ camera intalled on prismatic banister. The banister can can be rised for aproximately 20 cm so operator can see surrounding terrain and all nearby obstacles. During remote controlled operation an operator can control

Figure. 1: Photo of the RUDA robot.

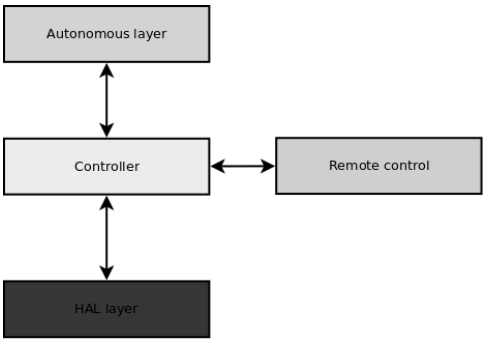

Figure. 2: Layers of control architecture.

robot tracks by joystick and receives visual feedback from cameras. The robot can be extended by several modules that bring additional sensors and effector in form of manipulator. All the modules are optional and robot can operate without them however some sensors are necessary for autonomous mode of robot (described later).

Internally robot uses hierarchical architecture as described in figure 2. Basic control loops are managed by simple fast modules. High-level and more complex tasks are implemented as proceses on universal main computer with operating system and high level of abstraction from hardware. Software on the main computer compounds of three layers: a HAL layer, a controller layer and an autonomous layer. The HAL layer is responsible for low-level communication with hardware control modules and other hardware. The HAL provides abstraction from particular buses and low-level protocols for layers above it. Robot hardware is interconnected with the main computer via CAN bus, RS-232 bus or via ethernet. HAL provides single entrypoint with unified interface for communication with particular hardware components.

The controller is a layer above HAL thas is responsible for transfer and convertion of control commands and feedback data. Controller is also responsible for communication with operator. Last but not least the controller implements control logic of the robot. It is responsible for avoiding run of conflicting tasks, for command prioritization and for interconnection of fast low-level control loops to get overall information about state and condition of the robot.

III. Available modules

The hull described in previous section is intended as a platform that can be extended with extension modules. Extention modules bring additional sensing and action capabilites for the robot. Currently modules include Manipulator, Bioradar, Avalanche finder and Sensor module.

A. Manipulator

Manipulator module adds a three degrees of freedom manipulator with two-finger gripper. It can be mounted to the rear module socket on robot hull. Manipulator can cary about 2kg of payload and it is equipped with additional camera with thermovision. It helps the operator to recognize living person under debris in cases when person can not be observed by normal camera. Operator can switch amongs normal camera and thermocamera to see the scene in both modes.

Manipulator can be controlled with two approaches: with controlling of particular joints and with endpoint possition control. In the first case each joint of the manipulator is completely controlled by operator. This mode is usable in cases when operator needs very precise control of the manipulator. In the second case the manipulator is controlled by setting endpoint position. Movement of the manipulator is computed, planned and executed by robot computer. This mode is easier to learn for the operator and more convenient. Disadvantage is that the manipulator does not see obstacles around itself so it can not be used in cases when manipulator needs to avoid obstacles while moving. Of course operator can switch between those two modes instantly. Gripper of the manipulator is controlled separately - controll of the gripper is the same for both modes of the manipulator.

Manipulator is equipped with two BLDC motors in shoulder and elbow joints and with one DC motor that rotates its base. The BLDC motors are controlled by dedicated hardware controllers that implement BLDC motor control algorithms [8]. The DC motor in manipulator base is controlled by manipulator HAL directly. HAL interface for the manipulator allows higher control layer to control manipulator movements with necessary safeguards (motor overload, hard joint limits) and provides those layers encoder data. The high-level control layer manages trajectory planning and collision avoidance during manipulator movements. The high-level control of the manipulator is based on OpenRAVE [9]. It works with manipulator and robot hull kinematics model so it can prevent collisions of manipulator with robot hull or collisions of manipulator with itself. For trajecotry planning the manipulator uses RRT algorithm [10] and for computation of inverse kinematics it uses analytic IKFast solver implemented in OpenRAVE. Model of the manipulator can be observed in figure 3.

B. Bioradar

Bioradar is a device that can detect living beings behind solid obstacles. It uses a micro Doppler effect to detect micro motion of human body like breathing or heartbeat [11]. It sends short sine wave pulses and receives reflected signal. If the transmitted pulse meets moving obstacle like human chest during breathing or human heart it slightly changes its frequency. According this small change the bioradar device can detect living person behind obstacle. The bioradar we use on a robot uses UWB (ultra wide band) signal in a large part of a spectrum to better overcome obstacles that attenuate signal on particular frequency [12]. Unfortunately the

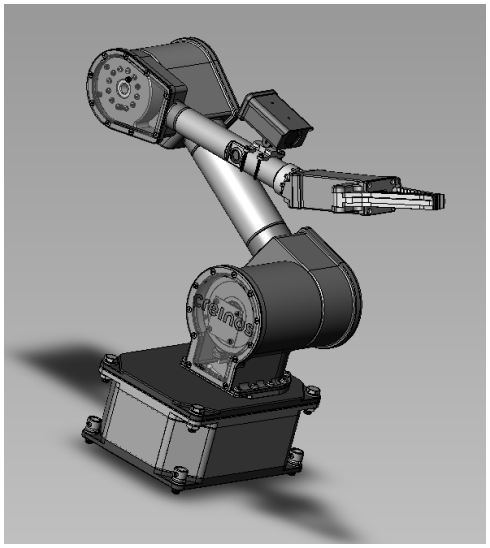

Figure. 3: Manipulator CAD model.

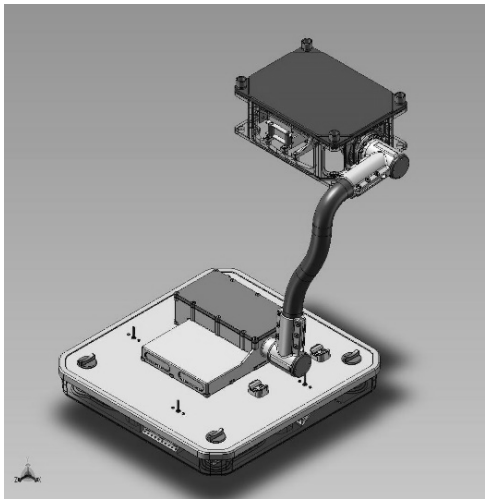

Figure. 4: Bioradar CAD model.

bioradar has several limitations. First of all it can detect only people with signs of life. If person does not breath and his/her heart does not beat than the person can not be detected by bioradar. Other problem is that the bioradar signal is intensively attenuated by metal. Many buildings use steel fittings in concrete constructions so some debris significantly decrease operational range of the bioradar. And last but not least the bioradar is tuned to reflect transmitted signal from human bodies that have similar properties as water so the bioradar does not work in cases the person is under water or snow avalanche. Still the bioradar is very usefull sensor for finding victims under debris or for finding hostile people hiding behind walls.
For use on RUDA robot the bioradar was integrated into bioradar module that can be observed on figue 4. The bioradar antena can po positioned by manipulator with two degrees of freedom. It can be placed on the floor or on the wall. For better placing the antena is equipped with four ultrasonic distance sensors that navigate the manipulator to place the antena as close as possible to the solid surface without colliding with it. These sensor allow robot to place the manipulator autonomously. This is used in autonomous mode described in chapter Autonomous Mode. The bioradar we used is a product of RETIA company. It was modified and integrated into bioradar module by our team.
The bioradar provides information about direction and distance of detected person in 3D space. Still output of the bioradar is difficult to interpret by machine due to false positives and uncertainity in its output. The bioradar output is displayed to the operator so operator can determine whether the person was detected or not. Automatic processing of bioradar output is a subject of further research.

C. Avalanche finder

The avalanche finder module is a simple module based on avalanche finder as shown on figure 5. The avalanache finder was extented with additional communication interface that allows HAL to read state of avalanche finder. The finder is always turned into receiver mode so it searches for signal from other avalanche finders that implicitly broadcast radio signal. The finder estimates direction and distance of broadcasted signal from receiver. It allows the robot to navigate closer to person under avalanche. Of cource avalanche finder requires the person to wear other finder. This is a limiting factor but usually if people go to avalanche field they cary avalanche finders.
The avalanche finder module can be mounted on rear socket of the robot. The finder itself is as far from robot motors as possible to avoid interference with motor electromagnetic field. From point of view of the operater the avalanche finder interface shows active detections and direction and estimated distance to detected signals. Output data from the finder are also used in autonomous modes when robot automatically searches area.

D. Sensor module

Sensor module is a module equipped with two cameras that work together as a stereocamera and LIDAR (a plane laser rangefinder) made by SICK co. This module has a driver connected to HAL that uses robots main computer to compute disparity for stereocamera. The LIDAR has a driver that only change format of data for higher layers of control system. The sensor module can be installed on front socket on robot hull - it can be installed on top of bioradar module too. A CAD drawing of the module can be observed in figure 6. During usage of bioradar the sensor module is covered by bioradar antena so data from sensors has to be ignored. Fortunatelly the robot can not place bioradar antena while moving so it is not limiting - robot has to home bioradar manipulator, activate sensor module and than it can move according to sensors again.
The module is essential for semi-autonomous and autonomous operation of the robot. Laser scans are used for construction of map of surrounding environment. This map allows robot to navigate autonomously to given target. The flat scan from laser is usually used in indoor environment. The stereocamera pointcloud on the other hand is usefull in outdoor environment where it helps the robot to get rid of terrain structure and obstacles. It provides information about obstacles that can not be seen by other sensors usual-

Figure. 5: Foto of avalanche finder used on the RUDA robot.

ly because they are above ground like tree branches or some hanging objects.

IV. Autonomous Mode

Despite the robot is primarily intended as remote controlled machine it is equipped with decision making system that allows the robot to work in partially or completely autonomous way. There are two main reasons for autonomous mode. First of them is to offload a part of work with control of the robot from operator to robot computer. It includes tasks like moving to designated target position, to search selected area with sensor or to avoid collisions and falls during high-level control. The second reason is recovery from situations when signal from operator station is lost. In such cases the robot tries

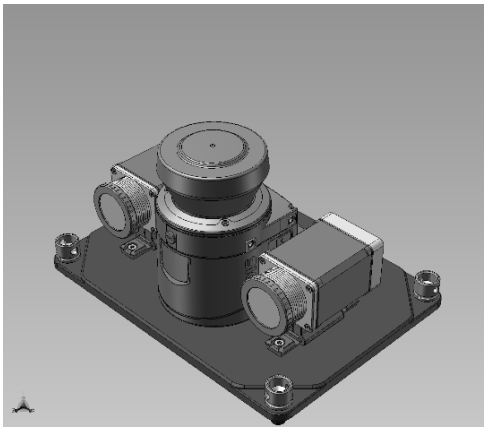

Figure. 6: Sensor module CAD model.

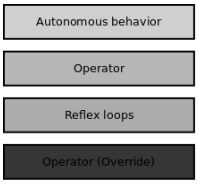

Figure. 7: Priority layers of control system.

to return back to place where it could receive signal from operator station last time.

A. Priorization of control layers

Entire control system of the robot is splitted into several layers according to priority. Prioprity layers make abstraction of control command priorization and fusion. The robot contains several sources of control commands: fast reflexive loops, autonomous behaviour algorithms and commands from the operator. Layers are stacked one on another according to its priority as shown in figure 7.
Under normal circumstances the top three layers are used during robot operation. The top layer represent command stream from the autonomous control system. On this layer come commands from trajectory following, autonomous mapping, area scanning and other autonomous tasks. These commands can be always override by actions of the operator. If operator decides to for example change trajectory of the robot, it is changed no matter what autonomous control system plans to do. Still command from the operator can be overrided. The highest priority have reflex loops. These loops include engine thermal protection, collision or fall avoidance according to ultrasonic sensors and malfunction signals and corresponding actions. This solution protects the robot from unintentional damage due to inadequate requests from the operator. Unfortunately the reflex loops are not infallible. If low-level protection mechanisms fail there is a possibility to send commands directly to HAL to allow controlling the robot in case of damage or partial malfunction. This is emeregncy situation and operator has to be aware of it. These emergency commands are represented by the bottom layer.

B. ROS

Autonomous control system of the robot is based on the Robotic Operating System (ROS) [13]. ROS is a middleware for developing robot controller with bundle of existing state of the art algorithms implementations. The ROS brings advantage of unified interfaces and data structures for data exchange between algorithm implemnetations. Main core component of ROS - the roscore takes care of delivering data between nodes - particular instances of binaries and scripts that implement given algorithms. In ROS the robot is modelled as a tree of solid objects that are interconnected with joints. Some of those objects are rather abstract like map or footprint of the robot base, other represent real components of the robot like tracks, sensors, etc.
For controlling the RUDA robot we use robot model with tree representation shown in figure 8.

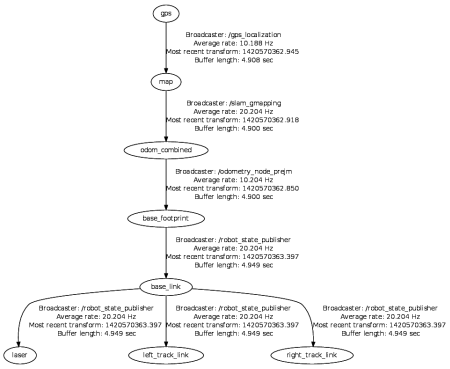

Figure. 8: Tree of transformation frames used in robot control based on ROS [13].

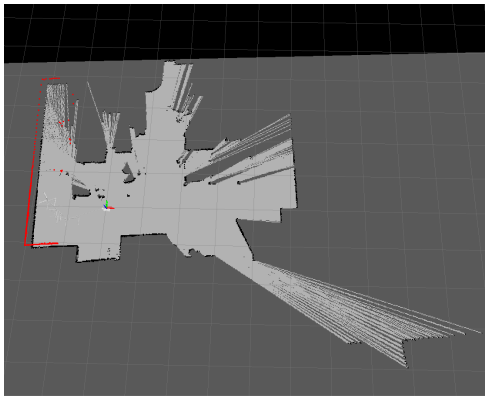

Figure. 9: Map generated by SLAM algorithm.

C. Sensors and data processing

The hull is equipped with several cameras and ultrasonic sensors. Moreover the robot can be extended with sensor modules including LIDAR, stereocamera, thermocamera and bioradar. Each those sensors provides data in different format with different properties and thus the data require different postprocessing.

Measured ranges from LIDAR are projected into flat plane into (x,y) coordinates according to measured distance, index of the sample and orientation of LIDAR sensor transformation frame to map transformation frame. Example of such projection can be seen on figure 9. Before the ranges are projected into plane they are filtered according to measured distance. If the distance is higher than threshold the measured distance is considered as ”infinite” - out of range of the sensor. It is due to growing error of measurement with growing distance from the sensor. There is also limit for too small distances. These ranges are ignored and not projected at all. The limit for dropping measurements is given by robot model - if measured point falls into robot body (with small threshlod) it is dropped.

The stereocamera sensor compounds of two cameras with synchronized timing. The driver of the sensor obtains synchronized data and computes diparity map. The disparity map is primary output from the stereocamera. The disparity data from stereocamera are converted into pointcloud and then the pointcloud is converted into virtual laser scan that is processed the same way as scans from LIDAR. This conversion is due to navigation system, that works in 2D. The 2D navigation is mostly sufficient for ground robot in indoor environments and is much less demanding for computational power.

Ultrasonic sensors measurements use filter for removing outliers. When measurement lies too far from centre of the cluster it is dropped. Other sensor data use rather simple processing including data format and unit conversion. These sensors include odometry, bioradar, avalanche finder, cameras and thermocamera. Bioradar, avalanche finder, theromocamera and cameras data are evaulated by operator.

D. Mapping and localization

When working with maps the robot controller may operate in two modes: SLAM mode when robot tries to build a map of unknown environment and localize itself in this map and localization mode when map of the environment is given (usually from prior SLAM walkthrough) and robot needs to localize itself in the map of environment according to its sensor observations.

In mapping and localization tasks we intesively use particle filter based algorithms as they are non-parametric, rather universal with decent performance. For SLAM mode we use existing implementation of Rao-Blackwellized particle filter SLAM available in ROS called Gmapping [6]. The SLAM process obtains data from odometry and scans from plane laser scanner (LIDAR usually) and contructs map accordig to these data.

Localization in given map uses adaptive Monte Carlo localization algorithm [4]. The algorithm is based on universal particle filter algorithm as described in [14]. Each particle represents particular pose of the robot. Enhancenment above universal particle filter algorithm is adaptive change of amount of particles placed in the environment according to density of particles in state space. We use existing implementation from ROS called AMCL that uses Augmented MCL and KLD-MCL algorithms [15].

E. Ongoing improvement of localization system

Improving the localization system is the main objective of ongoing research. Existing localization work well in indoor environments but it has sometimes poor performance in outdoor environment. When travelling long distances in flat terrain it is sufficient for the robot to use GPS. But for proper autonomous operation we need to localize the robot with better precision and also in situations when GPS signal is not available we need to be able to localize the robot.

Especially during change of surrounding environment (entering tunnel, driving from meadow to forrest or just going from indoor to outdoor environment) we need to change approach to localization to get good results. Change of approach include change of algorithms used and also change of utilized sensors. To switch between location sources we use system that selects localization source according to its ”trust in itself”. The trust could be expressed as covariance matrix or as probability. For localization source selection the equation

1 is used.

$$source = \underset{i}{\operatorname{argmax}}(\frac{t_i}{max(\vec{t})}) \quad (1)$$

This approach is used for absolute localization sources. Absolute means that it provides absolute position in map frame or world coordinates frame. On the other hand the relative localization based on odometry and IMU provides location relatively from starting position without known transformation to any global map frame. The transformation from odometry to map frame is given by absolute localization. When absolute location is obtained the begining of the odometry frame is projected into map frame. Also map frames need transformation from one to another. This transformation is obtained when system is localized in both map frames (world map frame for GPS and map generated by SLAM for laser-based navigation).

If we get location in several frames we need to align the frames and find transformation from all the frames to reference frame. First we transform robot pose p_{src} from source frame to reference frame as describes equation 2.

$$p_{ref} = T_{src\_ref} \cdot \vec{p_{src}} \quad (2)$$

Then in the reference frame the coordinates are alligned. It means that difference between pose of robot in reference frame is subtracted from transformed pose and new robot pose is computed called p_{ref}^{align}. This pose is transfomed back to source frame by multiplication with inverse tranformatin matrix (equation 3).

$$p_{src}^{align} = p_{ref}^{align} * T_{src\_ref}^{-1} \quad (3)$$

Finally the tranformation matrix from source to reference frame is updated using equation 4.

$$T_{src\_ref}^{new} = T_{src\_ref} \cdot find_T(p_{src}, p_{src}^{align}) \quad (4)$$

The alignment operation needs to be run periodicaly because of updates in robots location. Some location system might loose precision during robot operation or loose data completely but still it necessary to keep its frame aligned because of projection of points from one frame to another.

In our system the final position of the robot is obtained by fusing relative location system with absolute location system. From absolute location system the most trustworthy source of location data is used as described above and it is fused with relative location system - in our case with odometry. For fusing location data we use the Kalman filter algorithm in form desribed in [14].

Algorithm 1 Kalman Filter Algorithm (as presented in [14])

1: **function** KALMANFILTER($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
2: $\bar{\mu}_t = A_t\mu_{t-1} + B_tu_t$
3: $\bar{\Sigma}_t = A_t\Sigma_{t-1}{A_t}^T + R_t$
4: $K_t = \bar{\Sigma}_tC^T(C_t\bar{\Sigma}_tC_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$
6: $\Sigma_t = (I - K_tC_t)\bar{\Sigma}_t$ **return** μ_t, Σ_t
7: **end function**

The input of the filter if following: previous fused pose and covariance matrix μ_{t-1} and Σ_{t-1}, odometry as a feedback from the regulation action u_t and finally output from the absolute position system as measured observation z_t.

F. Approaches considered for localization

As mentioned before sensors used for localization in indoor environment is plane LIDAR optionaly fused with stereocamera. This solution works well indoor but it performs poorly outdoor - plane LIDAR and 2D map is unusable due to uneven terrain. It is not sufficient to distinquish between obstacles and free space - we need to consider if robot can drive over given place. This depends also on direction the robot comes to this place - for example robot can go down steeper hill that what it can climb up. Stereocamera theoretically provides depth map of the surrounding world but it has serious limitations and the data are too sparse for proper map building. Moreover simple geometric map of the terrain may not be sufficient for localization in cases robot is on large flat plane like meadow.

This is the reason we consider using other sensors and localization approaches in outdoor environment. First solution we consider is appearance based localization approach based on detecting known views or landmarks with camera - similar approach is described in [16]. This solution usually does not provide extremly good precision but it is reliable in point of view of probability that the recognized place is really the place the robot is located at.

Another approach is matching of 3D laser scans into map of environment as described in [17]. This solution can provide good estimation of real robot pose but in environments without geometric markers it can easily get lost. It is convenient to fuse both approaches to obtain more robust localization system.

Another way to go is using external sources of location information. It includes GNSS like GPS, Glonass, Galileo and also supportive sensor networks that provide localization information. Using GNSS is a standard in outdoor localization however precision of this localization approach is usually in meters. Precision can be significantly improved using differential GPS approach. Drawback of this solution is that mobile robot becomes dependent on stationary support system. This solution can not be used in some missions that robot is designed for (Searching avalanche field).

G. Trajectory planning

For trajectory planning we use existing well tested algorithms. Navigation system of the controller uses two levels of trajectory planning - the first one is global trajectory planning for planning trajectory from actual position to destination point and the second one is local trajectory planning that plans robot motion to avoid imminent collisions with obstacles. Local planning is used for following the global trajectory with actual robot pose and state.

For global planning the A* algorithm in occupancy grid is used. This algorithm provides good performance with acceptable computational demands. We use existing implementation of the algorith included in ROS Global planner package [18]. The algorithm takes into account already length of computed trajectory and also heuristic estimation of distance to the destination during computation. This provides

optimal trajectory while elliminating many useless state expansions compared to uninformed path planning algorithms. For local planning the Dynamic Window Approach algorithm is used. The algorithm is stochastic. It stochastically suggests values for forward and turning speed of the robot and simulates expected behaviour of the robot for particular parameters. It simulates large quantity of trajectories and for each trajectory it computes heuristics that say how the estimated trajectory fits the desired one. From all the simulated trajectories it takes the one that fits the best and uses its initial parameters. The algorithm is repeated every time the controller system sends update for robot base velocities. We use implementation of DWA algorithm existing in ROS [19].

H. Manipulator control

Manipulator controller works a a subsystem of central control system of the robot. Manipulator controller is responsible for trajectory planning, collision avoidance and for inverse kinematics computations. We use existing implementation of manipulator controller called OpenRAVE [9]. Advantage of this solution is flexibility when manipulator kinematic and dynamic parameters are modified. In comparison wiht other solution the OpenRAVE uses analytic inverse kinematics solver that has much better performance. For trajectory planning the stochastic BiRRT [10] algorithm based on Rapidly-exploring Random Tree algorithm is used. It tries to connect initial and final pose of the manipulator with randomly generated moves. For collision avoidance simple geometric intersection is used. Manipulator collision model compounds of geometric primitives - cylinders, cubes and spheres so collision computation is not extremly demanding.

I. Decision making

Decision making system is responsible for switching between high-level tasks according to situation and operators requests. Decision making algorithm is represented by a hierarchical state machine depicted in figure 10. After robot boots and high-level control system is started it jumps into green "START" state. From this state robot control jumps immediately into "OPERATOR CONTROLLED" state. This is default beahavior - the robot waits for operator actions. In this state the operator can start particular mission which leads into activation of particular superstate with corresponding caption. All the superstates share one superstate for trajectory following. Trajectory following with obstacle avoidance is the most utilized high-level beahvior. It is essential for most of robots missions.

The simplest mission superstate is used for searching avalanche fields. When mission starts the robot expects operater to designate polygon on real world map that will be searched. After robot controller receives polygon vertices it plans trajectory and jumps into trajectory following superstate. In this state controller simply follows trajectory point to point. When trajectory is interrupted by unexpected obstacle it plans avoidance trajectory and iserts it into pending trajectory. Then it continues following updated trajectory point by point. During search of avalanche field the trajectory following could be interrupted by sensor detection - when avalanche finder detects signal from avalanche broadcaster. In this moment controller informs the operator and records position of detected signal for future analysis. The mission ends when robot travels through the entire designated polygon. After end of mission the controller jumps back to "OPERATOR CONTROLLED" state.

Exploration of debris and removal of explosives are missions that mostly rely on operator. The controlller takes care of supportive tasks like approaching to area of interest when mission starts and mapping during operation in interesting area. When mission starts the controller computes approach trajectory to reach interesting area and follows this trajectory autonomously. This state can be always interrupted by operator action or it will end automatically when robot reaches desired position. In this moment controller activates mapping and hands over the control of the robot to the operator. Mission ends when operator terminates it. At this moment controller stores created map and deactivates mapping. Then it jumps back into default global "OPERATOR CONTROLLED" state. During mission when robot is controlled by operator signal from operator station could be lost. In this case controller jumps into "COMPUTE RETURN TRAJECTORY" state, plans return trajectory and start to follow it until communication is restored. If communication can not be restored, controller stays in "COMPUTE RETURN TRAJECTORY" state and waits until robot will be rescued.

Probably the most complicated superstate is used for searching of contamined area mission. If this superstate is entered the controller actiaves mapping and jumps into "OPERATOR CONTROLLED" state where it waits for operator commands. In this state operator can drive robot through the area. When sensor detects person controller notifies the operator and records position of the detection. Operator can swithc to autonomous mode when robot searches some area autonomously. If this action is started, controller jumps into "WAIT FOR POLYGON" state and waits for receiving informations about polygon of interest. Then it plans trajectory for searching the polygon and starts to follow it. The trajectory following can be interrupted by sensor detection. In this case the detection is stored and trajectory following continues. If operator performs any action the controller jumps back to "OPERATOR CONTROLLED" state. From this state the operator can terminate the mission. In such case the controller stores created map and deactivates the mapping. Finally it jumps back to the default global "OPERATOR CONTROLLED" state.

V. Missions and Use Cases of The Robot

The RUDA robot was designed to be able to solve various missions in various environments. Mission vary in complexity of solved tasks, in environment and operation conditions and also in posibility of using autonomous behaviour. Robot operation time is limited to cca 30 minutes. This time vary according to modules installed on the robot and usage of those modules and also according to intensity of robot motion. Unfortunately the robot was not tested in real missions - all experiments were conducted in artificial situations in testing polygons or during demonstrations. Still the robot proved itself to be able to solve particular tasks required during real missions so it is supposed to be usable for real missions. The 30 minutes long operation time is sufficient for following missions.

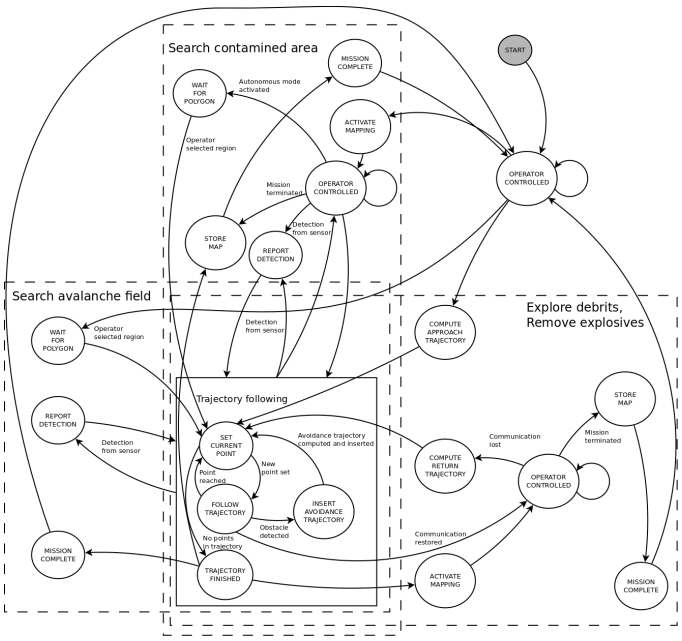

Figure. 10: Decision making state machine.

A. Searching debris

In case of earthquake or accident the building might collapse. The robot can drive through the debris and search for surviors. It can overcome difficult terrain in debris and it can use bioradar and thermocamera for searching. In this mode the robot has to be controlled by operator most of the time. Autonomous mode can be used for moving towards the area with debris. Robot can use manipulator for removing small obstacles and for checking statics of debris before humans enter the area.

B. Searching contamined area

Robot can operate in areas contamined by dangerous chemical or biological pollution. The hull and most of modules are prepared for chemical decontamination. Robot surface is also heat resistant so the robot can withstand fire for a short period of time. It can use camera, bioradar and thermocamera during search for surviors in this area. In this kind of mission robot can operate in autonomous mode and it can make map of area from covered with data from thermocamera an bioradar. Operator can go through the data after robot finishes the mapping. Unfortunately some parts of the robot including tracks, bioradar antena and LIDAR can not be decontamined so they have to be disposed after mission and replaced by new parts.

C. Searching avalanche field

For operation on snow the robot has to be equipped with wide tracks. Robot can search given area according to GPS autonomously. In autonomous mode the robot can avoid obstacles and report if it detects signal from another avalanche finder. In case of falling into the snow the robot still reports its position so it can be recovered later.

D. Removing explosives and other dangerous objects

Another use case of the robot is manipulation with dangerous or suspicious objects. Suspicios object can be bag left somewhere on public place. It is dangerous for human to approach the bag because it may contain explosives. It is safer to use remote controlled robot to reach the bag, grasp it a move it to the pyrotechnic container or to another safer place. Robot can also manipulate with dangerous chemicals and thanks to good surface protection it can operate in environments that are inacessible for human. Robot is not armored so it can not withstand explosion in its proximity but still sacrifice of robot is less serious than sacrifice of human life.

VI. Conclusion

The paper introduces the project of rescue robot with acronym RUDA and described some interesting parts of entire robot. Currently the robot can be used in several use cases. Greatest advantage of the robot is that it can take the risk of operating in dangerous areas instead of humans. Robot is remote operated however it has limited autonomous capabilities. Unfortunately the robot itself is a prototype that is not used in real missions but the robot was presented to rescue teams from several areas including fire brigade, pyrotechnists or police teams and they were interested in testing this prototype. The robot was also awarded by gold medal on International Engineering Fair (MSV2015) [20].

Despite the robot can be used in real missions there are many things that could be improved. There is a space for improving robot's chassis to improve transmittance through rough terrain. With additional stabilizating rear wheel the robot could climb up greater terrain gradient . Better dumping would help the robot to overcome difficult terrain where it has to fall from height. This is a common situation when riding over debris. Replacing manipulator with stronger one would allow robot to cary pyrotechnic water beam or shotgun for dismantling bombs. Stronger manipulator in general would allow the robot to remove bigger obstacles during its operations.

Another area of improvements is sensoric system of the robot. We received several suggestions about additional sensors and effectors that robot could be equipped with. The suggested sensor include roentgen scanner that is used in pyrotechnists missions to scan suspicious objects, sensors of chemical pollutions and also sensors for measuring stability of damaged buildings. Other sensor would help the robot to move more smoothly and effectively - for example pressure sensors on gripper fingers or optical-flow sensor for improving dead-reckoning.

Great challenge is improving and extending the autonomous mode of the robot. Mission control state machines could be extended to handle unexpected situations more autonomously. There is also a lot of space for improvements in localization of the robot as mentioned above in the paper.

To conclude the RUDA robot has still space for improvements and further development but in actual state it is an interesting solution for rescue teams in several mission types. Potential of the robot will propably grow with improvement of chassis and autonomous control and also with additional sensor and effector modules.

Acknowledgment

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Reliability and Security in IT project (FIT-S-14-2486

References

[1] "Laboratory of telepresence and robotics - project orpheus." http://www.orpheus-project.cz/products/. Accessed: 2015-10-20.

[2] "teodor explosive ordnance (eod) robot." http://www.cobham.com/about-cobham/mission-systems/unmanned-systems/products-and-services/remote-controlled-robotic-solutions/teodor-explosive-ordnance-%28eod%29-robot.aspx. Accessed: 2015-10-15.

[3] K. O. R. Haraguchi, S. Makita, "The development of the mobile inspection robot for rescue activity, moira2," *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics*, pp. 498 – 505, 2005.

[4] D. F. Patrick Pfaff, Wolfram Burgard, "Robust monte-carlo localization using adaptive likelihood models," cit. 2016-1-3.

[5] S. L. B. Joel A Hesch, Dimitrios G Kottas and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research January 2014*, pp. 182–201, 2014.

[6] W. B. Giorgio Grisetti, Cyrill Stachniss, "Openslam - gmapping." http://openslam.org/gmapping.html/. cit. 2015-10-10.

[7] S. G. Junnan Song, "Slam based shape adaptive coverage control using autonomous vehicles," *System of Systems Engineering Conference (SoSE), 2015 10th*, pp. 268–273, 2015.

[8] L. Zaplatlek, "zen bezkartovch (bldc) motor.," Master's thesis, Univerzita Pardubice, Fakluta elektrotechniky a informatiky. Pardubice, 2009. cit. 2013-04-25.

[9] R. Diankov, "Automated construction of robotic manipulation programs," August 2010. Carnegie Mellon University, Robotics Institute, CMU-RI-TR-10-29, http://www.programmingvision.com/rosen_diankov_thesis.pdf.

[10] D. F. R. Diankov, N. Rattlif, "Bispace planning: Concurrent multi-space exploration," *Robotics: Science and Systems IV*, vol. 63, June 2008.

[11] B. K. J.S. Kim, M. Rahman, "Dsp embeded hardware for non-contact bio-radar heart and respiration rate monitoring system.," *FBIE 2009, BioMedical Information Engineering*, p. 560563, december 2009.

[12] a. RETIA, "An introduction to doppler spectrum: Retwis application note.," 2012.

[13] C. of authors, "Robotic operating system." http://www.ros.org/. cit. 2015-09-23.

[14] D. F. Sebastian Thrun, Wolfram Burgard, *Probabilistic robotics*. MIT Press, Cambridge, Massachusetts, England, 2005.

[15] N. Berg, "Amcl ros package." http://wiki.ros.org/amcl. cit. 2016-1-1.

[16] A. Kelly, "Mobile robot localization from large scale appearance mosaics," *IEEE Transactions on Robotics and Automation*, 2000.

[17] J. G. Tingting Liu, Wei Zhang *et al.*, "A laser radar based mobile robot localization method," *Proceeding of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013.

[18] J. Lee, "Ros documentation: global_planner." http://wiki.ros.org/global_planner. cit. 2016-4-3., updated 2014-10-21.

[19] S. T. Dieter Fox, Wolfram Burgard, "The dynamic window approach to collision avoidance." http://www.cs.washington.edu/ai/Mobile_Robotics/postscripts/colli-ieee.ps.gz. cit. 2016-1-3.

[20] "International engineering fair - awarded exhibits gold medal 2015." http://www.bvv.cz/en/msv/msv-2015/gold-medal/awarded-exhibits/. Accessed: 2015-10-18.

Author Biographies

Radim LUŽA was born in 1987. He received his MSc. degree at FIT BUT in 2011. His research is aimed on robotics and localization systems.

Martin DRAHANSKÝ was born in 1978. He received his MSc. degree at FEI BUT (Czech Republic) and simultaneously at FE FernUniversitt in Hagen (Germany) in 2001. He obtained his Ph.D. degree at FIT BUT (Czech Republic) in 2005 and the position associate professor in 2009. His research interests include biometric systems, sensorics and IT security.

František V. Zbořil was born on 10. 11. 1945. He is an associate professor of Computer Science at the Faculty of Information Technology, Brno University of Technology, Czech Republic. He received his Ph.D. in 1978 at the same university. He started his research activities on analogue and hybrid computers with simulation of continuous systems. His next research was focused on classical artificial intelligence, robotics and neural networks. Now, the main objects of his professional interests are soft computing problems. He is the author of more than 100 papers and he is a member of several educational, research and academic boards or societies.