DR 2.3: Autonomous navigation, exploration and manipulation

Petter Ögren[*], Luigi Freda[†], Mario Gianni[†], Kalle Knipp, Rainer Worst[‡], Renaud Dube, Abel Gawel[¶], Tomáš Svoboda[‖] and the TRADR Consortium

[*]*KTH - Royal Institute of Technology, Stockholm, Sweden*

[†]*Alcor Laboratory, Department of Computer, Control, and Management Engineering "Antonio Ruberti" - Sapienza University of Rome, Italy*

[‡]*Fraunhofer IAIS, Sankt Augustin, Germany*

[¶]*ETH Zürich - Autonomous Systems Lab, Zurich, Switzerland*

[‖]*CTU - Czech Technical University in Prague, Czech Republic*

⟨petter@kth.se⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2013 (50 months)
Due date of deliverable:	December 2016
Actual submission date:	February 2017
Lead partner:	KTH
Revision:	Final
Dissemination level:	PU

This document describes the work of providing the TRADR robots with motion capabilities. For the UGVs we have investigated network aware navigation and path planning, traversal of smoke filled areas, mobile manipulation, mapping that handles dynamic areas with moving objects, improved general path planning, and improved tracking of planned trajectories. For the UAVs we have investigated indoor SLAM.

Executive Summary

This report describes work towards providing the TRADR robots, Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) with motion capabilities, in terms of mobile manipulation and navigation between different parts of the mission area.

During the third year, WP2 included tasks T2.3 (Semi-autonomous and autonomous mobile manipulation) and T2.6 (Autonomous navigation and exploration) and targets milestone MS2.3 (Autonomous navigation, exploration and manipulation).

In task T2.3 (Semi-autonomous and autonomous mobile manipulation), we have created a new control mode for investigating objects of interest, inspired by CAD software interfaces. We have also continued our work on network aware teleoperation, with a user study showing that the new interface with signal strength gradient information does indeed lead to more objects found and less lost connections than a standard interface displaying only signal strength.

Task T2.6 (Autonomous navigation and exploration) included work on both UGV and UAV. For the UGVs we have enabled the robots to traverse smoke filled areas, where both camera and laser scanners are blind, by using the flippers as tactile sensors. We have also developed mapping techniques that are able to adapt to rapidly changing environments and filter out objects that were moved. We have enabled improvements between sorties by enabling the loading and use of old maps for new subsequent missions. The path planning is made on both global and local levels, where the global level can take estimated network signal strength information into account. Finally, we have used learning techniques to improve the UGV tracking of a planned trajectory. For the UAVs we have added indoor SLAM capabilities using two orthogonally mounted laser scanners.

Role of navigation, exploration and manipulation in TRADR

Mobility of the TRADR robots is of key importance to the successful execution of the disaster response scenarios. In many instances, the overall system performance is improved when both operator and robot can contribute with their key strengths.

Contribution to the TRADR scenarios and prototypes

The motion capabilities of the TRADR UAVs and UGVs are essential for the use cases. In particular, the content of this deliverable relates to the following use cases

- Generic use case 1: *UAV[x] detect/search for X, using method Y* Capabilities for this use case are described in Section 1.5 below.
- Generic use case 3: *UGV[x] go to location X (optionally via Y).* Capabilities for this use case are described in Section 1.3 below.
- Generic use case 4: *UGV[x] go to location X on (semi)autonomous mode.* Capabilities for this use case are described in Section 1.3 and 1.4 below.
- Generic use case 5: *UGV[x] detect/search for X, using method Y.* Capabilities for this use case are described in Section 1.4 below.
- Generic use case 6: *UGV[x] manipulates object X* Capabilities for this use case are are described in Section 1.3 below.
- Generic use case 7: *UGV[x] encounters obstacle X, takes action Y to overcome* Capabilities for this use case are described in Section 1.3 and 1.4 below.
- Generic use case 8: *UGV[x] avoids colliding with actor X* Capabilities for this use case are described in Section 1.3 and 1.4 below.

1 Tasks, objectives, results

1.1 Planned work

The work described in this report (D2.3) was performed within the scope of Tasks T2.3 (Semi-autonomous and autonomous mobile manipulation) and T2.6 (Autonomous navigation and exploration), targeting the milestone MS2.3 (Autonomous navigation, exploration and manipulation). The objectives of these tasks were given as follows:

- *The goal of Task 2.3 is to develop Semi-autonomous and Autonomous modes for mobile manipulation.*
- *The goal of Task 2.6 is to provide new algorithms to allow UGVs and UAVs to collaboratively plan their path in order to navigate, explore, and monitor a disaster area.*

1.2 Addressing reviewers' comments

Below we collect the reviewer comments made in Year 2 regarding WP2, with corresponding answers.

1. Overall recommendations WP2 (part 1): the excessive planning time issues have been investigated and reported to be resolved. However, the unexpectedly large planning time has again been observed in the demonstration. **Response:** The problem has been solved by using different strategies which improve the algorithmic efficiency of the path planner and hence its response time (see Sections 1.4.5 and 1.4.6)
2. Overall recommendations WP2 (part 2): Regarding approaches for intelligent teleoperation and mixed initiatives, it needs to be made clearer how they go beyond the SoA. **Response:** See item 6 below.
3. Collision avoidance seemed not well developed yet for the UGV. Therefore, adding a lower-level control layer in the UGV to avoid collisions with obstacles based on LIDAR data and override higher-level commands coming either from autonomous path planning or human teleoperation that might put in danger the robot and its surroundings is of utmost importance and must be addressed as soon as possible by the research team in Y3. **Response:** A laser proximity checker has been added in order to check obstacle proximity with an high update rate. This information is used both by the traversability node and the trajectory control node (see Section 1.4.9).
4. The teleoperation interface presented in the 1st day of the review meeting that provides the user with preferable moving directions with respect to network connectivity is very interesting and useful in real

scenarios. **Response:** The work on network connectivity aware control has continued, as suggested by the reviewers. See Section 1.3.

5. The efficacy of the Free Look Control (FLC) method in non-flat and rough terrain is yet questionable. Although interesting results in flat terrain teleoperation were presented in the deliverable D2.2 and in the 1st day of the review meeting, these results cannot easily be extrapolated for non-flat and rough terrain where the TRADR UGVs usually operate. **Response:** A video demonstrating the successful performance of the FLC in rough terrain will be showed at the review meeting.

6. The review of literature in Deliverable D2.2 is too narrow and does not cite sufficiently representative relevant recent work on shared control and mixed initiative planning, e.g. in the scope of the recent DARPA Robotics Challenge but also in other contexts. It must be updated as furthermore the youngest reference being cited for intelligent teleoperation and shared control of UGVs in Sect. 1.6.1 is from 2009 and in Sect. 1.6.3 on mixed initiative planning is from 2010. Therefore, it has not been made sufficiently clear, where the approaches for intelligent teleoperation and mixed initiative planning really go beyond the state of the art. **Response:** The work on shared control was published at SSRR 2016 [8]. The reason for not including DRC references was that the approach is applicable to tracked UGVs only, and none of the papers on DRC robots found were using tracks. Finally, the paper does contain newer references, but the text in D2.2 unfortunately did not. Regarding mixed initiative planning, there were effectively newer research works that have not been included neither in D2.2 nor in the included annex. However, this work has been delayed (for the main reasons documented in Deliverable DR4.3) to focus the research work in Year 3 toward extending the path planning algorithm, as described in Section 1.4, and integrating it into the TRADR overall system.

1.3 Task 2.3 Semi-autonomous and autonomous mobile manipulation

In this section we will describe both the work on mobile manipulation, and the continued work on network aware teleoperation.

1.3.1 Mobile Manipulation

Using a Mobile Manipulator to explore an object of interest in a scene is anologous to exploring a virtual 3D object using 3D design software. In this work we make use of these similarities to propose a new control interface for

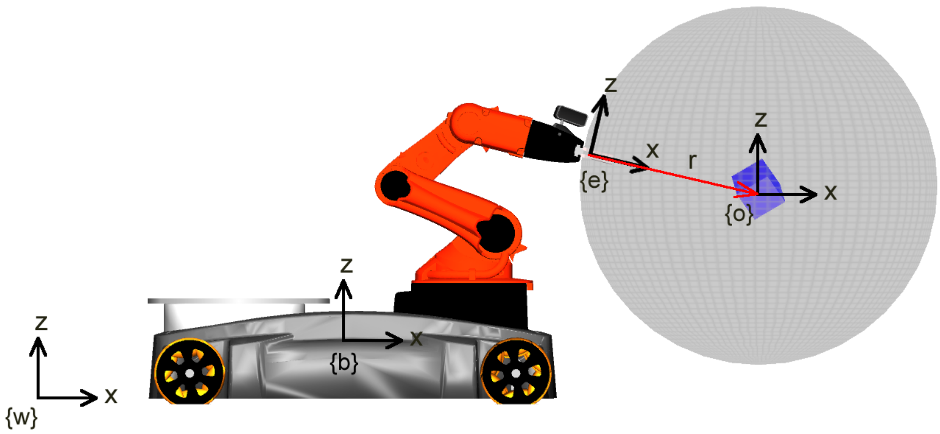


Figure 1: Control commands are interpreted in a spherical coordinate system centered at $\{o\}$ for *orbit object*, while keeping the x-axis of the end effector frame $\{e\}$ aiming towards $\{o\}$, and z-axis pointing upwards, as shown above.

teleoperated mobile manipulators. Even though the activities are similar, the user interfaces are quite different, mainly for historical reasons.

3D design software, such as Autodesk AutoCAD, SolidWorks, V-REP[1] and Gazebo[1] are used by engineers and architects to make 3D drawings and designs. When manipulating objects, the users navigate the virtual space using the functions *pan object* and *orbit object*. These two functions are the core part of an interface that is used daily by thousands of professionals and has been refined in many iterations. Therefore, there is reason to believe that the same functions would be useful when exploring and manipulating remote objects with a teleoperated mobile manipulator equipped with a depth sensor, such as an RGB-D camera[2].

With the proposed approach, the user can choose between *pan object* and *orbit object* when controlling the robot. In both control modes, the full pose of the end effector, position and orientation is controlled using a gamepad, and the video stream from the camera mounted on the wrist of the end effector is shown to the user.

In *pan object*, the sensor equipped end effector moves in a so-called robot centric way, known from the literature, see e.g. [6]. A requested translation forwards results in the end effector moving on a straight line in the direction towards whatever is in the center of view, and a requested translation to the right results in a straight line to the right.

The second mode, *orbit object*, has no counter part in robotics, and correspond to commands being interpreted in a spherical coordinate system

[1]The last two examples are actually robot simulators, but the functionality still concerns creating virtual 3D environments

[2]Red Green Blue plus Depth (RGB-D), with an image including depth/distance information, such as the Microsoft Kinect, Intel RealSense and PrimeSense.

(ϕ, θ, r) centered on the object of interest, as shown in Fig. 1. Command right now means orbiting the object in the ϕ-direction, up means moving towards observing the object from above in the θ-direction, and forwards means moving towards the object in the negative r-direction. It is interesting to note that the well known singularities of polar coordinate systems have not presented a problem to users.

The two control modes described above complement each other, and are often used in an alternating fashion. However, as *pan object* is equivalent with robot centric control, [6], we focus this work on *orbit object* which is new to the robotics community.

In the design applications, *orbit object* is useful for exploring an object, and moving into a viewpoint that allow you to add or remove details. In the robot teleoperation application, we believe that *orbit object* would be useful for exploring objects, getting good camera views from all sides. It would also be useful when gathering 3D data, in order to create a high quality 3D-model. Finally, it would be convenient when deciding on the appropriate grasp point for lifting an object, or operating a door handle.

The approach presented in this paper realizes the *orbit object* control mode using constraint based programming. To realize the appropriate motion of the end effector, the configuration of the complete mobile manipulator must be taken into account. Sometimes it is best to move only the arm, but sometimes the mobile base has to be moved to increase the range of the arm, while simultaneously taking obstacles and internal singularities into account. All this has to happen automatically, enabling the user to focus on the task at hand, which is being carried out by moving the end effector relative to the object of interest. For details, see the appended paper [9] (Annex Overview 2.1).

1.3.2 Network Aware Teleoperation, Design and User Study

Recent and current Urban Search and Rescue (USAR) missions show that the range and coverage of the wireless connection between the operator and the teleoperated Unmanned Ground Vehicle (UGV) presents a significant constraint on the mission execution. For continuing operation, the operator needs to continuously adapt to the dynamic network connectivity across the environment in addition to performing the primary navigation, observation and manipulation tasks.

In this work, a new teleoperation User Interface (UI) is presented that integrates information on the Direction of Arrival (DoA) of the radio signal. The proposed approach consists of (1) a method for estimating the DoA and (2) a color-bar representation surrounding the video feed that informs the operator which navigation directions of motion are safe, even when moving in regions close to the connectivity threshold.

The UI was evaluated in a user study with 24 participants who performed

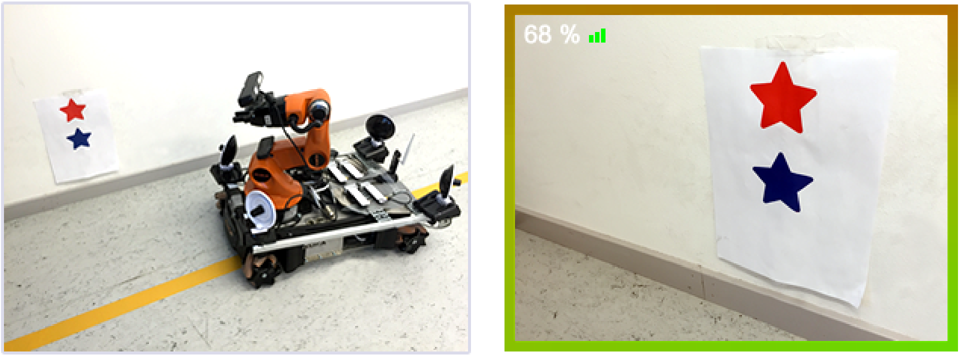


Figure 2: The youBot mobile robot equipped with wireless network hardware used in the experiments (left) shown along with the user interface (UI) displaying the RSS DoA as a color bar around the video feed from the robot.

a search task under challenging wireless connectivity conditions. The results show that using the proposed interface resulted in more objects found, and less missions aborted due to connectivity problems, as compared to a standard interface.

Today, teleoperated UGVs play an increasingly important role in a number of high risk applications, including USAR and Explosive Ordinance Disposal (EOD). The successful completion of these missions depend on a reliable communication link between operator and UGV, but unfortunately experiences from Fukushima and the World Trade Center disaster show that cables can limit performance, or break [80], and wireless network connectivity can be lost [77].

Despite improvements in wireless technology, it is reasonable to believe that the very nature of USAR scenarios imply a high risk of damages to infrastructure, including electricity and network facilities. To avoid relying on wireless technology, one possible solution would be to enable the UGVs to operate autonomously, but for the foreseeable future, human operators will remain more versatile than autonomous systems when it comes to decision making, in particular in challenging and unpredictable USAR environments [105, 79]. Therefore, Connectivity awareness is viewed as a component of Situation Awareness (SA), determining where the robot can operate.

In this work, we address the problem of improving SA such that the operator is aware of dynamic network connectivity and adjust the UGV operation to it. This is done by extending the user interface (UI) with not only a measure of Radio Signal Strength (RSS), but also a notion of the motion direction (i.e. the DoA) that would increase this signal strength, and thereby the communication quality (delay, packet loss, etc.) which has shown to affect teleoperation task performance [90].

Using the proposed solution, an operator close to the connectivity limit knows which way to go to improve the connection. An operator who, for example, would like to move the UGV a bit more to the left to inspect a cavity, knows if this move will improve, worsen or leave the RSS unchanged.

The proposed UI is composed of two parts, first the DoA is estimated, then it is presented to the operator in an efficient manner.

The estimation of the DoA is done by using spatially dispersed wireless receivers on the four edges of the UGV (as can be seen in Fig. 2) and applying the finite differences method to extract the RSS gradient. We then employ spatial and temporal filtering schemes to mitigate multipath fading effects and transient noises in the measurements. The estimation and filtering algorithms run online and dynamically adapts to the wireless environment such as a change in network connection (e.g. introduction of an intermediate relay robot as a signal repeater) or a mobile wireless access point connecting the robot to the base station.

The presentation of the DoA to the operator was chosen in view of the fact that gaining a good SA is very challenging in USAR missions [69]. In fact, it was shown in [16, 111] that as much as 49% of mission time is normally devoted to improving the operator SA. Further, it was recommended in [112] to use a large central part of the screen for the video feed. Therefore, we propose to add the DoA information in the form of a color bar surrounding the video feed (see Fig. 2) to provide SA to the operator in terms of network connectivity and physical surroundings.

For the evaluation, we identified two important challenges associated with teleoperation of UGVs in USAR missions: (1) providing effective SA to the operator and (2) ensuring resilient wireless connectivity with the UGV. High SA can reduce mission time and improve operator decisions, while a resilient network connection will avoid losing control of the UGV. For details, see the appended paper, Section 2.2.

1.4 Task 2.6 part 1: Autonomous navigation and exploration using UGVs

1.4.1 UGV blind traversal/exploration

When the UGV enters an area with dense smoke both Lidar and camera sensors go blind. We designed an algorithm [43] (Annex Overview 2.4) for a sequential stop-and-go blind terrain exploration using the flippers, see Fig 3 for an overview. The basic loop works as follows:

1. Set flippers to desired shape.
2. Advance (blindly) forward a certain distance (20 cm).
3. Lower the flippers to perform tactile sensing (or touch) with the front flippers in order to estimate the height profile of the terrain.

It should be noted however, that not only terrain profile is important. Also the actual state of the robot, e.g. whether it is currently climbing up or

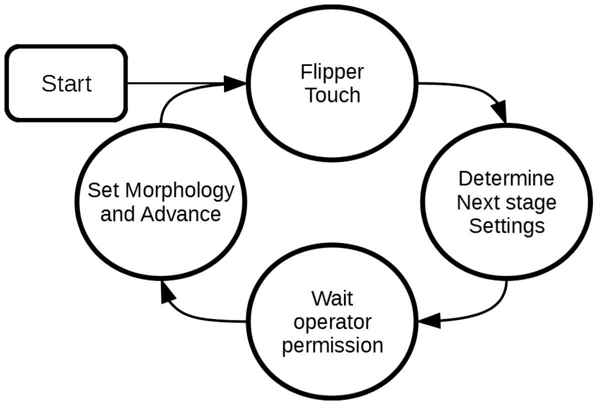


Figure 3: Blind exploration algorithm, basic scheme. The *wait operator permission* is optional.

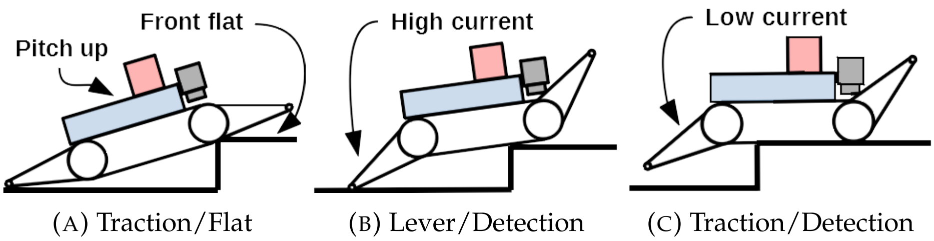


Figure 4: A strategy for climbing-up blindly.

down. To determine the state, other interoceptive measures are used. Several strategies are proposes, as illustrated in Fig. 4 for an example.

We also designed an algorithm for the robotic arm to explore space the in front of a mobile robotic platform to identify obstacles that could damage it [15] (Annex Overview 2.5). The algorithm moves the robotic arm in the environment, measuring terrain features and obstacles by touching them.

1.4.2 Autonomous navigation and exploration

Dynamics within a sortie The TRADR system has the objective of assisting and collaborating with the human search-and-rescue team as early as possible and at the place in need. During the first response, it is likely that the place is presenting changes while the robots are exploring the area. These changes can be structural or caused by the movement of victims. Thus, the on-line mapping module has to account for these *fast* changes in order to obtain a robust and reliable model of the environment.

To this end, the TRADR mapping module has been improved by adopting the *OctoMap* representation, instead of the transformed 3D point cloud, as described in [56]. This representation is efficient as it uses an *OctTree*

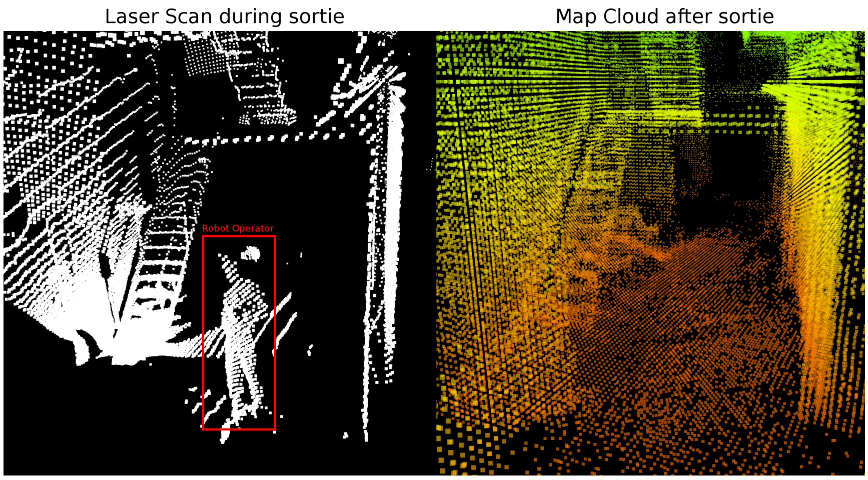


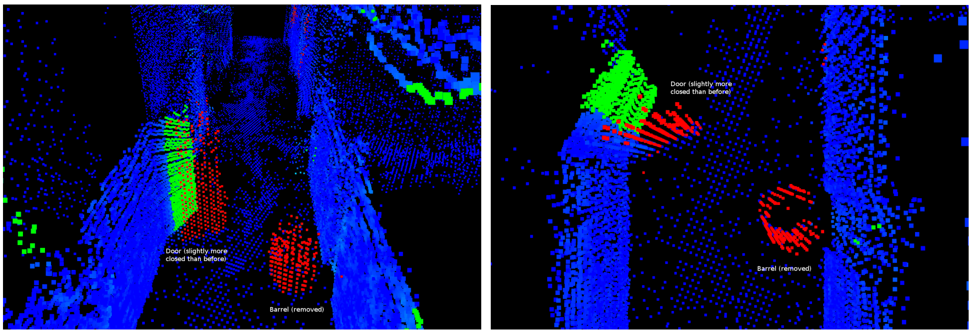
Figure 5: Fast dynamic filtering using the *OctoMap* representation: A human present in a LiDAR scan (left) is filtered out of the scene (right).

as a data structure, where every 3D voxel contains the confidence of being occupied or not. In this way, when there are returns on dynamic objects, the confidence of being occupied for the corresponding voxels increases. Given that the voxels does not generate returns anymore this confidence goes down, declaring the voxels as free. An example of this filtering process can be seen in Fig. 5.

Dynamics between sorties With the passage of time, more sorties are performed by the UGVs. The environment models built during different sorties can then differ. Those changes are of special interest for the first-responders as they could indicate locations of high risk of structural-damage.

Our approach to detect dynamic elements between sorties is closely related to the work of Girardeau-Montaut et al. [50], who propose to identify changes based on a distance threshold, computed on every observation in a reference point cloud to its closest neighbor in a newly acquired point cloud. However, since real world applications usually consist of partially overlapping missions, this point-cloud based approach becomes infeasible, as non-overlapping areas have a high distance in a purely geometric sense. In contrast to [50], we also handle observed and un-observed portions of an environment, by employing the *OctoMap* representation which mitigates the drawback of a point-cloud based approach in the case of only partially overlapping map segments.

Our procedure takes an assembled laser scan, *i.e.*, a 360° depth image and aligns it to a previous sortie's *OctoMap* using the robot's current state estimate. The distance comparison is then performed only on points that have been observed before. Previously unobserved points are initially



(a) (b)

Figure 6: Example of detected changes between sorties: A door was moved between sorties and a barrel disappeared. Sideview (a) and topview (b).

treated as static. Finally, we apply clustering based on spatial density on the points of high distance and apply a machine learning algorithm to classify the clusters as either actual dynamics or outliers.

In Fig. 6 we show an excerpt of the changes detected and highlighted between two sorties.

Map loading and alignment As mentioned above, one of the key advantages of the TRADR system is its ability to exploit the persistence over time of the maps built through different sorties. We have implemented the key capability of loading the previously recorded and already optimized maps in the TRADR system for use in the next sortie. This map loading, alignment and use for path-planing, as demonstrated in TRADR TJEX 2016, has been reported in [32].

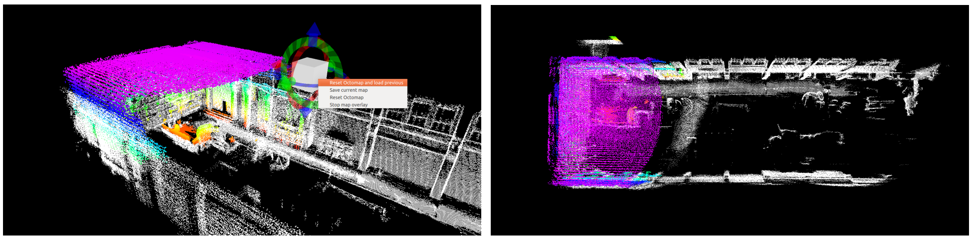
Furthermore, we developed a plugin for visually handling loaded maps and map overlay to allow map loading in offset locations. The loaded map can also be accessed by the path planner for autonomous functionalities. Finally, it enables a consistent transformation tree for multi-robot applications and the usage of previously recorded maps across different UGVs, as shown in Fig. 7.

1.4.3 Mapping and Point Cloud Segmentation

At each new scan, a structure interpretation of the map is updated. In particular, the map point cloud is segmented in order to estimate a pointwise traversability of the terrain.

In a first step, the point cloud map is built and filtered using the Octomap representation described in Sect. 1.4.2.

Next, geometric features such as surface normals and principal curvatures are computed and organized in histogram distributions. Clustering is



(a) (b)

Figure 7: Loading of previous map in an offset location, the map can interactively be moved for visual alignment.

applied on 3D-coordinates of points, mean surface curvatures and normal directions [73]. As a result, a classification of the environment in regions such as *walls*, *terrain*, *surmountable obstacles* and *stairs/ramps* is obtained.

The segmentation can be computed both on a saved map, which can be loaded by the user in the octomap server, or on a live map, which is incrementally built during a mission. Note that the normals computation are more efficient and reliable if a saved trajectory is available for the current map[3].

1.4.4 Traversability Cost

Traversability is computed as a cost function on the built map by taking into account the described point cloud classification and the computed local geometric features [37]. In particular, the point-wise traversability cost function $trav : \mathbb{R}^3 \longmapsto \mathbb{R}$ is defined as

$$trav(\mathbf{p}) = w_L(\mathbf{p})(w_{Cl}(\mathbf{p}) + w_{Dn}(\mathbf{p}) + w_{Rg}(\mathbf{p})) \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ is a generic Euclidean point, the weight $w_L(\mathbf{p})$ depends on the point classification, $w_{Cl}(\mathbf{p})$ is the cost contribution provided by the clearance of the robot w.r.t. surrounding obstacles and other robots of the team, $w_{Dn}(\mathbf{p})$ is a function of the local point cloud density and $w_{Rg}(\mathbf{p})$ measures the terrain roughness (average distance of outlier points from a local fitting plane).

A *traversable map* is then built by suitably thresholding the clearance cost $w_{Cl}(\mathbf{p})$ and collecting the resulting points along with their traversability cost value.

In principle, once the traversability map is built, this could be stored and used in subsequent missions. We decided not to follow this strategy since the current map can be heavily readjusted by the SLAM module and

[3] At present time, a map can be saved along with the corresponding robot trajectory.

in that case the segmentation must be done from scratch. Thus, currently traversability can be computed from a saved map.

1.4.5 Global and Local Path Planners

Path planning is performed both on global and local scales. Given a set of waypoints as input, the *global* path planner is in charge of (1) checking the existence of a traversable path joining them and (2) minimizing a *mixed cost function* along the computed path (more details in Sect. 1.4.7). This mixed cost function combines together the traversability cost (see Sect. 1.4.4) along with a task-dependent cost function, which can be used for instance to guide the robot where the WIFI radio signal strength is higher.

Once a solution is found, the *local* path planner safely drives the robot towards the closest waypoint by continuously replanning a feasible path in a local robot neighbourhood in order to more readily take into account possible dynamic changes of the environment.

Both the global and the local path planners capture the connectivity of the configuration space $\mathcal{C}$ by using a sampling-based approach. For efficiency reasons[4], the path planner computes trajectories for the robot directly on the traversability map by internally representing the robot body with its bounding sphere (i.e. the smallest sphere containing the robot). This allows the path planner to restrict the path search to a projection of $\mathcal{C}$ on a 3D Euclidean space. A tree is expanded on the traversability map by using a randomized A* approach [37]. The start and goal nodes are computed as the projections of the start and goal robot positions on the traversability map. At each step,

- the *safe radius* $R_{\boldsymbol{n}}$ of the current node $\boldsymbol{n}$ is computed as the minimum of the clearance of the robot from obstacles at the position corresponding to $\boldsymbol{n}$ and a maximum allowed robot step;
- a set $\mathcal{N}$ of neighbours of $\boldsymbol{n}$ is created by collecting all the traversable points that fall within a safe ball centered at $\boldsymbol{n}$ with radius $R_{\boldsymbol{n}}$;
- a small percentage of neighbours in $\mathcal{N}$ are randomly selected as children with a probability inversely proportional to the traversability cost (so as to bias the expansion towards more traversable regions);
- the A* cost-to-go of each new child is computed by taking into account the mixed cost function and is then used for inserting with priority the new child in the A* search queue;
- the next node to expand is extracted from the updated priority search queue as the element with the minimum cost-to-go.

[4] Note that the path planner must run on the robot main board and share computational resources with other demanding processing nodes.

In this process, all the visited nodes are efficiently stored in a kd-tree. The algorithm ends when a child node is found close enough to the desired goal position.

Once a saved map is loaded back in the system, a path can be planned on the corresponding computed traversability map.

1.4.6 Path Planning Windowed Search Strategy

In order to increase the efficiency and improve the response time of both the local and global path planners, a windowed search strategy has been implemented around the basic path planner. Let $\mathcal{S}$ be the Euclidean line segment between the assigned start and goal positions. Each time the global/local path planner is called to compute a new path:

- first, a path is searched only in the subset of points of the traversable map which are contained in a narrow box $\mathcal{R}_1$ whose median axis contains $\mathcal{S}$[5];
- if a path cannot be found in $\mathcal{R}_1$ then it is searched in a new region $\mathcal{R}_2$ which is built by suitably growing $\mathcal{R}_1$ along its axes of symmetry;
- if the path search fails this process is repeated by incrementally growing the search region until a maximum number of attempts is reached.

Clearly, in order to preserve the probabilistic completeness of the basic path planning algorithm, the last attempt uses the full traversable map as search region. For sake of safety, after each failed planning attempt, the last updated traversability map is considered as input.

This strategy proved to work very well in practice. Most times the path is found at the first step with the advantage of conveniently reducing the search space (the path planner only considers the most interesting and useful part of the traversability map). Moreover, this clearly reduces the actual computational costs of the path planner.

1.4.7 Mixed Cost Function

The randomized A* algorithm computes a semi-optimal path $\{\boldsymbol{n}_i\}_{i=0}^{N}$ in the configuration space $\mathcal{C}$ by minimizing the total cost

$$J = \sum_{i=1}^{N} c(\boldsymbol{n}_{i-1}, \boldsymbol{n}_i) \tag{2}$$

where $\boldsymbol{n}_0$ and $\boldsymbol{n}_N$ are respectively the start and the goal, and $\boldsymbol{n}_i \in \mathcal{C}$. The cost function $c : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}$ can combine traversability and a task-dependent

[5] Roughly speaking, this region is shaped as a corridor whose longitudinal axis is aligned with the line segment $\mathcal{S}$.

function. In particular

$$c(\boldsymbol{n}_i, \boldsymbol{n}_{i+1}) = (d(\boldsymbol{n}_i, \boldsymbol{n}_{i+1}) + h(\boldsymbol{n}_{i+1}, \boldsymbol{n}_N))\pi_1(\boldsymbol{n}_{i+1})\pi_2(\boldsymbol{n}_{i+1}) \tag{3}$$

$$\pi_1(\boldsymbol{n}) = \lambda_t \frac{trav(\boldsymbol{n}) - trav_{min}}{trav_{max} - trav_{min} + \varepsilon} + 1 \tag{4}$$

where $d : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}^+$ is a distance metric, $h : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}^+$ is a goal heuristic, $\pi_1 : \mathcal{C} \longmapsto \mathbb{R}^+$ is the normalized traversability function, $\lambda_t \in \mathbb{R}^+$ is a scalar positive weight, ε is a small quantity which prevents division by zero and $\pi_2 : \mathcal{C} \longmapsto \mathbb{R}^+$ is a normalized task-dependent cost function. The first factor in eq. (3) sums together the distance metric and the heuristic function (usually the distance to the goal). The other two factors π_1, and π_2 respectively represent a normalized traversability cost and a normalized task-dependent cost, whose strengths can be adjusted by using the weight λ_t. Note that $\pi_i \geq 1$. Clearly, a normalized task-dependent function π_2 can be built in a similar way.

1.4.8 Resilient communication-aware motion planning

Assuming the availability of a point-wise estimate of the Radio Signal Strength (RSS) $rss : \mathcal{C} \longmapsto \mathbb{R}$, the normalized task-dependent cost function π_2 can be defined as follows

$$\pi_2(\boldsymbol{n}) = \lambda_r \alpha_r e^{-t/\tau} \frac{rss_{max} - rss(\boldsymbol{n})}{rss_{max} - rss_{min} + \varepsilon} + 1 \tag{5}$$

where $\lambda_r \in \mathbb{R}^+$ is a trade-off scalar positive weight, $\alpha_r \in [0, 1]$ is a normalized estimation confidence and τ is an exponential decay constant (it determines the amount of time after which π_2 goes to its minimum value 1). The exponential decay is used to decrease the effect of the RSS cost after a certain amount of time (e.g. before the path planner is stopped by a timeout in a case solution is difficult to found). In this context, the RSS can be for instance estimated using a Gaussian Process Regression as described in Sect. 2.7.

1.4.9 Laser proximity checker and dynamic obstacle negotiation

In previous TRADR experiments, one of the main issues experienced during robot navigation was caused by the low update rate of the assembled 3D laser scans, as the robot has to wait 3-4 seconds in order to get a new 3D laser scan[6]. During this time the robot has to blindly rely on the built map in order to compute its navigation path. While this is not an issue if

[6]This is approximately the time required by the laser rolling motor to cover 180°. Moreover, in order to have a new 3D laser scan completely integrated/merged in the octree-based map, an additional time is needed by the laser mapper to register the new 3D scan with the reference point cloud map.

the full map of a static environment is available beforehand and the robot localization is perfect, this does not work in practice when the robot navigates in real rescue scenarios where dynamic changes are to be expected and localization accuracy can temporarily decrease.

In order to improve the robot navigation without the need of additional hardware, we designed and implemented an obstacle proximity checker node. This continuously collects and assembles together planar laser scans with an approximate update rate of 25 Hz, in a very similar way to the presently available laser scan assembler. The novelty here is (1) the parallel construction of an *obstacle point cloud*, which results from collecting (w.r.t. the base link frame) and filtering all the points which fall within a certain safety region, and (2) the computation of the distance to the closest obstacle point. Both the obstacle point cloud and the closest obstacle distance are emitted as ROS topics with the same update rate of the planar laser scan.

The obstacle point cloud is received by the traversability node which uses it in order to compute the robot obstacle clearance at each point of the traversable map. The result of these computations is then used by the local path planner in order to search a path that more quickly adapt to dynamic changes.

Furthermore, the closest obstacle distance is received by the trajectory control which significantly decrease the cruise velocity when obstacles are too close.

1.4.10 Second order inverse dynamics learning

The trajectory tracking controller of the TRADR UGV relies on a prior model of the inverse kinematics of the robot [46, 47]. This model neglects the non-linearities induced by the track-soil interaction, thus leading to inaccurate motion generation [47, 45]. This, in turn, leads to unsafe navigation, especially under unexpected dynamic changes of environment [36].

To cope with this problem a hybrid approach to trajectory control design is proposed. This approach relies on both a derived nominal model of the inverse dynamics of the TRADR UGV and a Bayesian non-parametric data-driven estimation, based on Conditional Independent Gaussian Processes (CI-GPs). The derived model serves to identify the principled variables of the control law underlying the trajectory controller as well as to drive the process of acquisition of the data. On the other hand, the CI-GPs have the main role of capturing all the dynamic effects which have not been explicitly accounted for in the second-order nominal model as well as of suitably managing uncertainty and noise of the measurements.

Unlike goal-directed techniques, where a model of the discrepancy between the nominal model and the real behavior of the robot is learned [84], the proposed hybrid method treats the nominal model as a sort of partial prior knowledge informing the system about the leading physical variables.

It also enriches the feedback provided by the nominal model with the estimation from observation of the required quantities in so also capturing unforeseen variability due to unmodeled dynamic effects, uncertainty and noise. Furthermore, it leverages the feedback error induced by the superimposition of the effects of the two models when combined together, taking also into account possible conflicts and misalignment between the individual estimates.

Parameter estimation for the CI-GP has been performed on data collected by giving as input to the trajectory tracking controller several reference trajectories with different shapes, inclinations and terrain surface frictions. After the estimation of the parameters of the CI-GP, we also grounded the model within the robot control schema to measure its performance along new planned trajectories, under loop closure. For circular trajectory references, the hybrid controller achieves an average error of ~ 0.1 m. In the presence of additional forces induced by the tilted reference trajectories, the controller achieves an average error of ~ 0.2 m. Finally on a reference path lying on a metal staircase the average error of the controller is ~ 0.02 m. More details about this work can be found in Section 2.8.

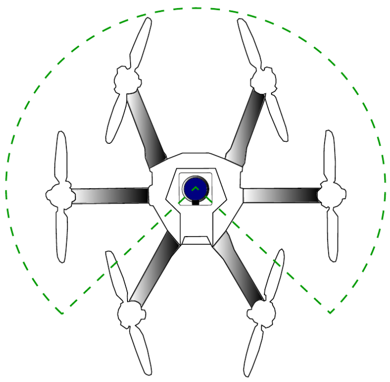
1.5 Task 2.6 part 2: Autonomous navigation and exploration using UAVs

Collaborative planning of mobile robots is only achievable if each robot knows about its own position and the positions of other robots, e.g., in order to merge a map created by a UAV with a map of a UGV. This section describes our approach to determine the position of a UAV depending on all six dimensions of freedom. Simultaneously, a three-dimensional map of the environment can be created. This approach was developed primarily for indoor environments, where often only a bad or no GPS Signal is available. The resulting solution to localize a UAV without using GPS is described in a thesis [60].

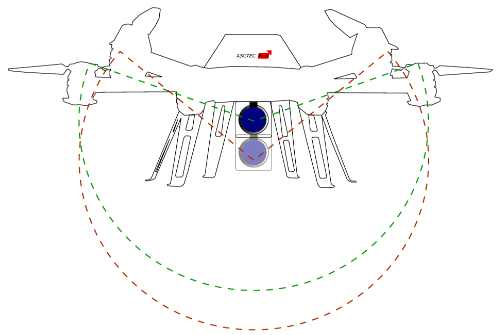
An overview of state of the art techniques for localization and mapping with laser scanners is given in the state of the art section of this document, Section 1.6.8.

1.5.1 System design

Hector SLAM is used to determine the position of a UAV depending on all six dimensions of freedom. This approach is particularly well suited because it includes the tilt information of the UAV's IMU. However, Hector SLAM supports only the creation of two-dimensional maps and the localization in a two-dimensional space. It is possible to extend this approach by using two laser scanners simultaneously. With this extension a three-dimensional



a) Laser scanner mounted in horizontal direction



b) Laser scanner mounted in vertical direction

Figure 8: Installation of two laser scanners on a UAV

map can be created and the position of the UAV depending on all six dimensions of freedom can be determined. Firstly, both laser scanners need to be mounted in the right way to detect position changes in the x, y, and z-direction. To achieve this, one scanner is mounted on the top of the UAV to capture the horizontal plane, see Figure 8 a. As shown in figure 8 b, the second scanner is mounted on the bottom to capture the vertical plane. When mounting both scanners, some issues, such as shading effects, need to be taken into consideration. Therefore, the laser scanner on the bottom side shouldn't be mounted too close to the UAV; the scanning range is extended by installing the scanner on the bottom side with a little distance.

The laser scanners used for localization and mapping are of the type UST-20LX from Hokuyo. These scanners have a low mass of 130 g and a high update rate of 40 Hz. A distance up to 20 meters can be measured and the scan angle is 270 degrees. Ethernet with 100 Mbit/s is used for data transfer [55].

The Neo from Ascending Technologies is used to carry the two laser scanners and to develop our approach for 3D localization and mapping. Additional payload up to 2,0 kg can be mounted, but it is important to take into account that the required batteries already count as playload [5].

Thus, the complete set up of all hardware components consists of two laser scanners, Hokuyo UST-20LX, an UAV Ascending Technologies Neo, two mounting devices to assemble both laser scanners, and one network switch to connect the scanners to the UAV's mainboard. The entire set up is shown in Figure 9.

Figure 9: Complete set up of all hardware components

1.5.2 Using Hector SLAM with two laser scanners

Hector SLAM is used twice, separately for each laser scanner. With this configuration, two two-dimensional maps of the environment are created. One for the horizontal and one for the vertical plane. As shown in figure 10 the translations and rotations are determined from both maps and combined by a ROS Node named *Position Matcher*. Missing position data are added from the UAV's IMU. This makes it possible to localize the UAV depending on all six dimensions of freedom. By creating a vertical 2D map of the environment, the altitude of the UAV can be determined more accurately than by measuring it directly with a distance sensor; it can be determined correctly over uneven ground, too. Also different air pressure has no effect like altimetry using an air pressure sensor.

1.5.3 Performance

A NUC Board with an i7-5557U Processor is used as the computing unit on the Neo. This Board is provided with two cores or four threads with a cycle frequency of 3.1 GHz and 8.0 GB RAM. Our approach for self-localization and 3D mapping was tested on this system and and required computing power is described below.

At idle mode, the CPU load varies between zero and four percent. Using the procedure described in section 1.5.2 the CPU load increases up to twelve percent and about 620 MB of RAM is used. Overall the system is not saturated by the provided technique and enough computing power is available for other applications. On the other hand it would be possible to use a computing unit that is lighter and has less power. This could have a

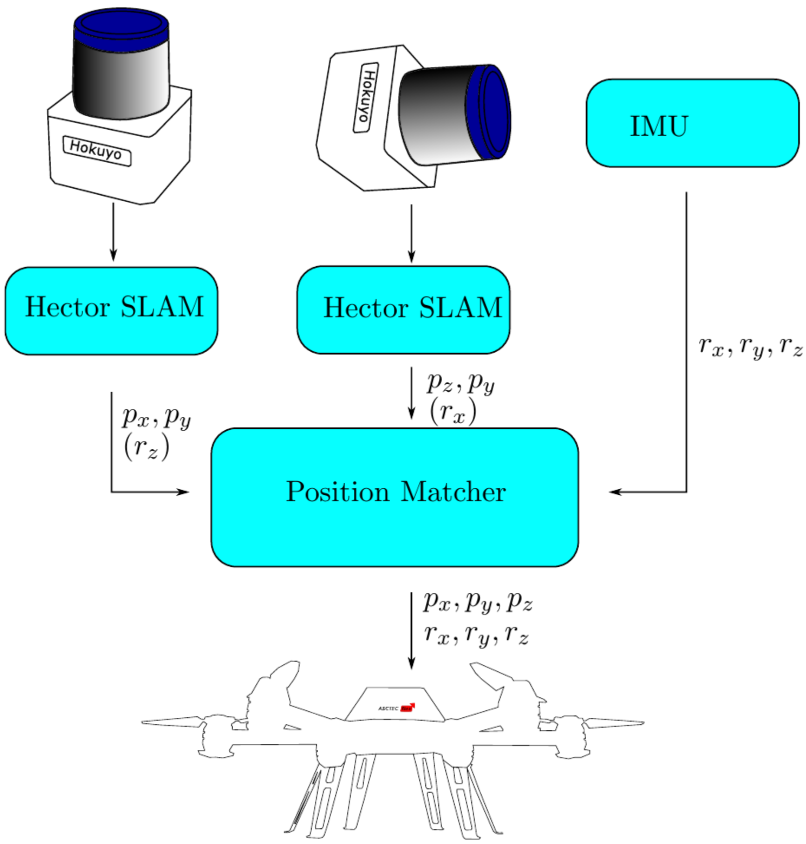


Figure 10: Fusion of all position data, to get the position of the UAV depending on all six degrees of freedom.

positive impact on the maximum flight time of the UAV.

1.5.4 Evaluation

To test the developed system in detail, a floor is chosen as test environment. A ramp is located at the end. Using our approach, the change in altitude is not measured directly, but with the help of Hector SLAM by creating a vertical 2D map. The ramp offers a good opportunity to check the correct estimation of altitude changes. The orientation of the laser scanner on the bottom side has a high influence on determining altitude changes correctly.

As shown in Figure 11 a, the bottom scanner is aligned to detect the yz-plane of the environment. The scanner cannot detect enough changes of the surroundings in this configuration, therefore a correct altimetry is not possible.

As shown in figure 11 b, a better result is given by rotating the laser scanner on the bottom side by 90 degrees. Now the vertical scan plane is

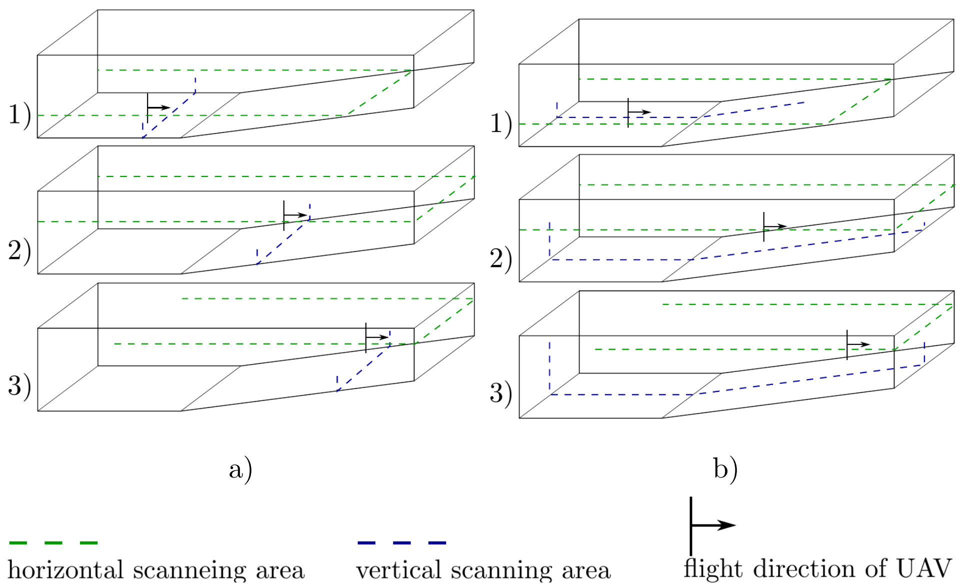


Figure 11: Influence of scanning area depending on the alignment of the laser scanner on the bottom side.

aligned to the flight direction and the scanner can detect enough changes in the environment to detect altitude changes. By flying straight forward the rotated position of the scanner does not support detection of the environment at the side of the UAV. So the ground of the test environment is not displayed very accurately. To get more information of the ground, special flight patterns can be executed, like climbing and descending, or rotations around the z-axis.

Additionally, the system was tested in other environments, e.g., at the TRADR Evaluation 2016 in Dortmund. A decommissioned power plant is available at this location. There were some possibilities to test the system in detail. Our technique for self localization and 3D mapping is primarily developed for indoor scenarios, so a hall inside the structure was chosen. Figure 12 shows laser-based 3D-models that were recorded indoor. The different shades of gray represent the intensity of the scanning points and are only used for a better optical recognition of edges and structures.

1.6 Relation to the state-of-the-art

In this section we will describe how the results of D2.3 relate to the state-of-the-art.

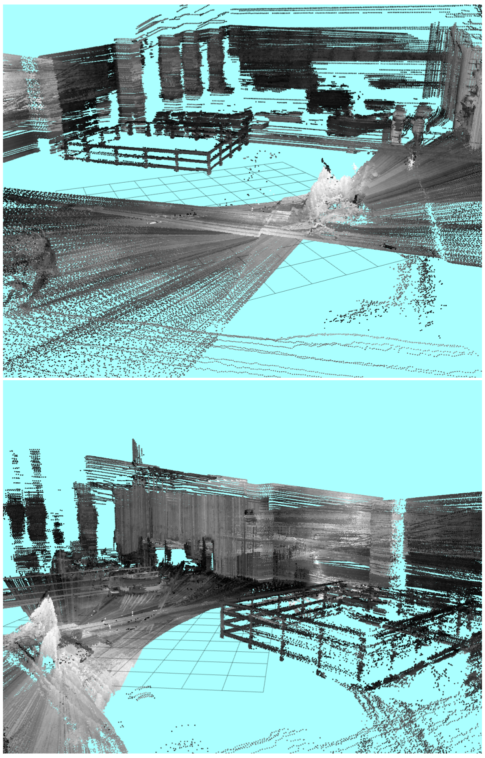


Figure 12: Power plant indoors - TEval 2016 Dortmund

1.6.1 Mobile Manipulation

The contribution of this work is that we show how to realize the control mode *orbit object* in a teleoperated mobile manipulator. To the best of our knowledge, this has not been done before. We also show how to incorporate avoidance of obstacles into the framework using constraint based programming.

Teleoperation of a 8 DoF mobile manipulator using a 6 DoF joystick was investigated in [41]. The authors propose a control approach where the user controls the gripper pose, while the mobile base adapts and follows the gripper when possible and needed, to avoid over extending the arm.

The approach proposed in this paper differs from the above in that the control mode is neither robot centric nor world centric, but object centric. In the *orbit object* control mode, a commanded motion to the right results in moving right with respect to the gripper, but keeping a constant distance to the object. This is motivated by an argument similar to the ones suggesting inspiration from computer game interfaces [92], but this time the inspiration comes from the professional 3D design community.

Motions can be constrained through virtual fixtures [94], and the approach used in this paper can be seen as such in the sense that regions can be restricted through constraints. However our implementation does not use force sensors for feedback, which is the case in for instance [11].

Introducing different constraints for restricting motion could introduce conflicts, in which case it could be necessary to prioritize [61]. In this paper we do not explicitly consider priorities, however by changing weights and introducing slack variables this could be considered.

1.6.2 Network Aware Teleoperation

The main contributions of this work are three-fold. We first propose a new way of estimating DoA for teleoperated UGVs. We then propose a way of integrating this DoA information in a UGV teleoperation UI. Lastly, we perform a user study, showing that the proposed approach in fact increases the number of found objects during a search mission, and decreases the chances of losing the connection to the UGV. To the best of our knowledge, none of these items have been done in a UGV teleoperation context before.

The wireless network connectivity of USAR UGVs have often proved unreliable [78, 19], with examples including real incidents where robots were lost during disaster inspection operations [80, 77]. Casper et al. [20] investigated user confidence in remotely operated robots with intermittent communications, and found that these problems had a significant impact on the usability of the systems. They even suggested that because of communication dropout problems, wireless robots should be avoided. However, the flexibility of wireless systems compared to tethered robots still make them an important alternative in many applications.

A natural way of avoiding loss of communications is to make the user aware of the connection quality. A decade ago, this information was usually not displayed in the Operator Control Unit (OCU) [40], but more recently, it is often added in the form of a "signal bar" (as in modern cell phones) or in form of a percentage. Typical examples of such representation can be seen in [68, 52] including the recent Quince 2 robot's OCU [114]. Furthermore, the Wayfarer OCU for Packbot robots [109] represent the radio signal level in a vertical bar manner, in addition to a numeric indicator.

The literature on robot interfaces also include examples where information about gradients and directions is made available to the user. In [54] two microphones on the left and right of the robot were used to estimate the direction of a sound source, which was displayed (overlaid on the video) in the form of a pointer floating on a horizontal line. A similar representation was used in [52] to show robot speed information. In [23], the authors proposed a tactile belt that vibrates in the direction of detected collisions to improve SA, while in [97] a study found that the use of a tactile vest did not improve SA significantly in navigation tasks.

An influential study in Human-Robot Interface (HRI) design [112] advocates the use of a large single interface with a significant percentage of the screen dedicated to video. The authors also recommend providing more spatial information about the environment to increase SA, and using fused sensor information to lower the cognitive load on user.

In this work we go beyond the related work described above by having the teleoperation interface include not only a scalar value to describe the network connectivity situation, but also the direction in which it is expected to improve, i.e. the DoA. Assessing the geographic distribution of network connectivity is a spatial task, for which the visual modality fits best with the human information processing (e.g., see the multi-resource model of Wickens [107]). Therefore we choose to present the DoA in the form of visual gradient bars surrounding the video feedback.

Carefully integrating the DoA information into the visual feedback is crucial. For this we use FLC (Free Look Control) [83] as the control layer. FLC is essentially a "navigate-by-camera" mode as envisioned in [113]. In the FLC mode, the operator controls the UGV in relation to the camera frame instead of the world frame, making it more intuitive than the traditional so-called *Tank Control* mode. Hence it is appropriate to use FLC for presenting the DoA information in direct reference to the camera frame, making the UGV control easier while simultaneously enhancing local SA. The proposed DoA interface integrated with FLC indeed has the ability to satisfy all the three levels of SA (perception, comprehension, and prediction/projection) mentioned in [33].

1.6.3 Blind terrain exploration

The task of the arm-based tactile exploration may by split into two main tasks. The 3D coverage path planning algorithm (CPP) [42] that plans the movement of the tactile end-effector and the actual arm motor-control. It is worth noting that most of the coverage path planning algorithms deals with a simple 2D (lawn-mover problem) [4] or 2D manifold in 3D (seabed inspection) [53]. We followed aheuristics based algorithms [96], [87] and extended it into 3D inspired by the idea of exploring consecutive planes [53]. For the arm control we combine inverse kinematics computation [10] for short movements and motion planning [70], [1] for larger movements. The blind traversal algorithm essentially builds on our previous work [104] and several hand-crafted heuristics tuned experimentally.

1.6.4 Dynamics detection in geometric data

Some research has been conducted on novelty detection in geometric data, which can be separated in two general types of approaches.

Approaches falling in the first category aim at detecting dynamics using individual points or voxels [7, 50, 38, 89]. These approaches generally achieve a high level of detail for the detected dynamics, because every single sensor measurements is used. This also enables the algorithms to detect objects disappearing from the scene. However, considering all measurement points requires to explicitly handle outliers in order to achieve robustness. The algorithms proposed by Azim and Aycard [7] and Pomerleau et al. [89] use multiple consecutive sensor measurements to deal with outliers and refine the detection. These techniques are however impractical for the TRADR UGVs due to the slow rate at which assembled point clouds become available.

The second type of approaches are those that cluster points and approximate these groups of points with a function. Novelties are then detected using these approximations and points are later re-associated with the functions to obtain detailed dynamics [30], [102], [3]. Because these approaches fit a function to a group of points they have a certain level of implicit robustness to outliers. The downside of this approximation is that information about the expected visibility of map points is lost. As a result, none of these approaches can detect disappearing objects.

1.6.5 Three-dimensional global and local path planning for the UGV

In the last decades, a vast amount of research work has been done in the field of robotics concerning three-dimensional motion planning for UGVs in unstructured outdoor environments [76, 75, 12, 21, 13, 73, 35, 14, 95, 67].

The aforementioned work highlighted that the main three key factors that have to be taken into account when dealing with motion planning problems are (i) the shape of the terrain [95], (ii) the physical properties of the robot [67] and (iii) the detection and recognition of dynamic obstacles [88, 36].

Morriset *et al.* in [75] proposed an approach for terrain traversability assessment based on a semantically annotated polygonal model of the terrain for the RHex mobile robot. Semantic labeling on registered point clouds is performed according to the geometrical properties of adjacent polygons. A multi-region planner chooses the most suitable planner, among a repertoire of primitives, based on the type of regions on which they are applicable. However, this approach is limited to the set of terrain types or structures which are imposed by the classification schema or to the set of planning primitives.

A graph-based algorithm searching for paths on a triangle mesh reconstruction of the environment is proposed in [12]. Given the discrete path along the triangle strip, a vector field for steering the robot from triangle to triangle is computed to obtain a smooth path over the meshes.

Brunner *et al.* in [14] developed a two-stage motion planning scheme for reconfigurable robots, similar to the TRADR UGV, which makes uses of a roughness quantification of the height-map of the environment. According to this schema, an initial path search on a motion graph is performed. This graph encodes information about areas of moderate roughness, steepness, regions of higher roughness and challenging steepness on the basis of the proposed roughness quantification of the map. The computed path, providing an approximate solution, is then used to initialize the RRT-based search for local expansion.

A method for assessing drivability from eight RGB-D sensors measuring the 3D terrain geometry is described in [95]. In this work, the authors aggregate omnidirectional depth measurements to robot-centric 2.5D omnidirectional height maps and compute navigation costs based on height differences on multiple scales. The resulting 2D local drivability map is used to plan cost-optimal paths to waypoints, which are provided by an allocentric terrain mapping and heuristic-based path planning method that relies on the measurements of the 3D laser scanner of the robot.

In [73] we developed a framework for 3D autonomous navigation for the TRADR UGV. In this framework path planning is performed on a graph-based representation of the environment. This representation was obtained from the semantic labeling of the point cloud, similar to [75, 21]. Points belonging to clusters, labeled as ground and stairs or ramps, on the basis of the estimated normal orientations, are connected via an iterative procedure taking into account both the model and the kinematic constraints of the robot, namely its morphology as well as its ability to overcome obstacles. The result of this procedure is a graph connecting the different regions of the point cloud, denoting areas accessible by the robot. In parallel, both boundary and inflated obstacle regions are estimated by projecting the points labeled as walls onto the planes tangent to the surfaces approximating ground, stairs or ramps. Upon the estimation of the boundary regions, the edges of the connectivity graph are weighted by a factor taking into account the distance of the graph vertexes from these boundaries, the density of the neighborhood of the vertexes and the arc length of the edge. This traversability structure is used by the graph-based planning strategy to find minimum cost feasible paths toward target goals.

In [34], the above framework has been improved under both terrain traversability cost estimation [14], and heuristic-search [28]. Further in [36] dynamic obstacle detection has been also taken into account.

Unlike the research work mentioned above, Krusi *et al.* in [67] recently proposed a method combining sampling-based planning and local trajectory optimization to compute system-compliant trajectories in the space of robot configurations directly on (unordered) 3D point clouds acquired with range sensors, avoiding any kind of supplementary terrain reconstruction process, such as computing a height or voxel map, or a mesh representation of the environment. In this method terrain assessment is only estimated locally and on demand during motion planning, based on the local distribution of points in the map.

1.6.6 Resilient communication-aware motion planning

Considerable efforts have been made to address the problem of maintaining a robust wireless communication between mobile robot(s) and the base station [93, 57, 81]. Most of the solutions focus on designing or using mobile repeater (relay) robots to establish and/or to repair an end-to-end communication link [74, 62, 85, 63]. Other solutions focus on means to provide situation awareness of wireless connectivity to the robot or the teleoperator [17].

An overview of the Communication-Aware Motion Planning (CAMP) problem is presented in [115]. Several works assume an oversimplified model in which the connectivity is modelled as a binary function. In this case, the predicted Signal to Noise Ratio (SNR) and the estimated distance of the robot (aerial or ground) to the radio source are empirically thresholded in order to identify regions with high probability of communication coverage. [58].

In [27], Authors propose an optimization strategy to compute a path along which the predicted communication quality is maximized. They made use of supervised learning techniques (Support Vector Regression) to learn and predict the link quality such as the Packet Reception Ratio. It is worth noting that in this case the learning mechanism is offline and hence it can only be applied to a static environment.

A communication aware path planner is proposed in [100] for an aerial robot. Here Authors present a probability function which is a function of the SNR between two nodes. The SNR model is learned from the measurements online using an Unscented Kalman Filter (UKF) model. Assuming a multi-hop network using aerial relay robots, a path optimizer that maximizes the probability of communication in all the intermediate links is proposed.

Works that combine communication and motion planning together are strongly influenced by Mostofi et al. In [72], Authors developed a mathematical framework to predict the communication quality (mainly the SNR) in unvisited locations by learning the wireless channels online. This prediction model is then used to define a motion planner either to improve channel assessment [44] or to optimize for communication and motion energy to reach a given target [110]. This framework is further extended in [2] to include online channel learning for co-optimization of communication transmission energy and motion energy costs. Here, the transmission power is modelled as a function of SNR, whereas the motion power is a function of robot's velocity and acceleration.

Recovering back from a communication failure is a topic that has not been much researched by the community. A simplistic solution could be to back-track the robot along the path it travelled as long as it regains communication. Alternatively, the robot can remember positions where the connection was lost and move the robot towards that position. In [26], a decentralized algorithm is proposed to move the disconnected robot towards a known position of the gateway (radio signal source or relay) by taking into account obstacles along the way. In [57], Authors demonstrated a behaviour to drive the disconnected robot towards the closest robot node (assuming a multi-robot network) and repeat this till connection is restored. Note that in above mentioned works the wireless channel parameters are not estimated but instead a perfect knowledge on the network topology is assumed (e.g, the positions of the gateway nodes, base station, etc.).

In the Wireless Sensor (or ad-hoc) Networks (WSN) community, where it is commonly assumed that ample amounts of hopping nodes are available, the problem of repairing a connectivity failure is viewed differently. In this case, mobile robots can be used as sensor nodes which can reposition in order to replace failed nodes [108] or can be introduced into an existing WSN for restoration [101]. A router repositioning algorithm (move towards node with weaker RSS) is proposed in [62], in order to restore connectivity in an ad-hoc network.

It can be seen that predicting the communication quality in regions not explored by a mobile robot is a challenging problem. As pointed before, probabilistic approaches such as maximum likelihood and UKF have been used to model the path loss and shadowing components of the RSS. Yet these models perform efficiently when there is at least some priori information on the network such as source or relay node positions, which is difficult to know in field robotic applications such as emergency deployment of robots to help in disaster response operations. In [39], a Gaussian Process based method is proposed to estimate the channel parameters and map the RSS in real-time using few sample measurements.

1.6.7 Inverse dynamics learning for embodied unmanned ground vehicles

Nowadays, machine learning techniques are commonly applied to significantly improve dynamic modeling of complex robots, especially of actively articulated tracked robots [47]. In this regard, a number of methods have been proposed combining contextual policy search (CPS) [25] with prior knowledge [22] and regression [82, 24]. CPS methods are typically based on local search based approaches such as Locally Weighted Projection Regression (LWPR) [103, 22].

A large portion of the literature also focused on employing kernel-based methods for the estimation of the inverse dynamics mapping by employing approaches, such as Gaussian Process Regression (GPR) and Support Vector Regression (SVR) [64]. Local Gaussian Process (LGP), introduced in [82], handles the problem of real-time learning by building local models on similar inputs, based on a distance metric and uses the Cholesky decomposition for incrementally updating the kernel matrix. In [49], the authors propose a real-time algorithm, dubbed SSGPR, which incrementally updates the model using GPR as learning method.

For the special case of relatively low-dimensional search spaces combined with an expensive cost function, global search approaches, like Bayesian optimization, have been proposed for selecting hyper-parameters [98]. Bayesian optimization has been used for non-contextual policy search in robot grasping [66] and for locomotion tasks [71, 18].

In [48] we proposed an approach for learning patterns of control manoeuvres for the TRADR UGV. In this approach the control policy of the UGV in encoded via a Dirichlet Process-Gaussian Process (DP-GP) mixture model. Gibbs sampling [91] and a hybrid Monte Carlo technique [31] are applied to obtain both an estimate of the concentrations and of the hyper-parameters of the model. A similar approach has been used for modeling non-linear dynamics of moving targets [59, 106].

1.6.8 Laser-based SLAM for UAVs

SLAM (Simultaneous Localization And Mapping) is a common approach to create a map of a mobile robot's environment. It describes the possibility to create a map of the environment while simultaneously localizing the robot within this map. No previous information about the environment is required for this approach.

Meanwhile many different SLAM implementations are available, based on different kinds of sensors like cameras or laser scanners [99]. *Structure from Motion* is often used by camera-based techniques to estimate distance information from an environment as an alternative to stereo vision. However, the most popular sensor option for SLAM are laser scanners. The distance of objects can be measured directly by laser scanners and does not need to be computed. So the computation effort can be reduced in comparison to image based techniques. Some implementations are available for laser-based SLAM in general. Implementations for ROS are already deployed for GMapping and Hector SLAM, which make it easier to use these applications on a robot. GMapping follows an approach based on particle filters, especially on the Rao-Blackwellized filter. Particles are used to estimate the hypothetical position of the robot [51]. The main advantages of Hector SLAM are the low demand of computing power and the high update rate of 40 Hz [65]. Hector SLAM can use position information from additional sensors like an IMU to improve the estimation of the robot's position. This makes the use of Hector SLAM on a UAV very attractive.

There are many different models of UAVs available including the DJI [29] or Parrot [86] to create maps of an environment. To mount additional sensor payload like laser scanners, a UAV is needed that allows a flexible setup. Ascending Technologies provides solutions that fulfill these requirements, especially the "Neo" model provides an appropriate platform to mount additional hardware.

References

[1] Ompl, open motion planning library. `http://ompl.kavrakilab.org/`. Accessed: 2016-01-06.

[2] Usman Ali, Hong Cai, Yasamin Mostofi, and Yorai Wardi. Motion and communication co-optimization with path planning and online channel estimation. *arXiv preprint arXiv:1603.01672*, 2016.

[3] Henrik Andreasson, Martin Magnusson, and Achim Lilienthal. Has somethong changed here? autonomous difference detection for security patrol robots. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3429–3435. IEEE, 2007.

[4] Esther M Arkin, Sándor P Fekete, and Joseph SB Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17(1):25–50, 2000.

[5] Ascending Technologies GmbH. Ascending Projektseite. URL `http://www.asctec.de/`. `http://www.asctec.de/` [Online: 17.03.2016].

[6] J A Atherton and M A Goodrich. Supporting Remote Manipulation with an Ecological Augmented Virtuality Interface. *(unknown)*, 2010.

[7] Asma Azim and Olivier Aycard. Detection, classification and tracking of moving objects in a 3d environment. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 802–807. IEEE, 2012.

[8] F. Båberg, S. Caccamo, N. Smets, M. Neerincx, and P. Ögren. Free look ugv teleoperation control tested in game environment: Enhanced performance and reduced workload. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 312–319, Oct 2016. doi: 10.1109/SSRR.2016.7784321.

[9] F. Båberg, Y. Wang, S. Caccamo, and P. Ögren. Adaptive object centered teleoperation control of a mobile manipulator. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 455–461, May 2016. doi: 10.1109/ICRA.2016.7487166.

[10] Patrick Beeson and Barrett Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 928–935. IEEE, 2015.

[11] Alessandro Bettini, Panadda Marayong, Samuel Lang, Allison M Okamura, and Gregory D Hager. Vision-assisted control for manipulation using virtual fixtures. *Robotics, IEEE Transactions on*, 20(6):953–966, 2004.

[12] A. Breitenmoser and R. Siegwart. Surface reconstruction and path planning for industrial inspection with a climbing robot. In *CARPI*, pages 22–27, 2012.

[13] Michael Brunner, Bernd Brggemann, and Dirk Schulz. Rough terrain motion planning for actuated, tracked robots. In *Lecture Notes in Computer Science*, pages –. Springer, 2013.

[14] Michael Brunner, Bernd Brüggemann, and Dirk Schulz. *Rough Terrain Motion Planning for Actuated, Tracked Robots*, pages 40–61. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-662-44440-5. doi: 10.1007/978-3-662-44440-5_3.

[15] Martin Burian. Haptic terrain exploration with robotic arm. Master's thesis, Czech Technical University in Prague, Czech Republic, 2017.

[16] J. Burke, RR Murphy, M. Coovert, and D. Riddle. Moonlight in Miami: An ethnographic study of human-robot interaction in USAR. *Human-Computer Interaction, special issue on Human-Robot Interaction*, 19:1–2, 2004.

[17] Sergio Caccamo, Ramviyas Parasuraman, Fredrik Båberg, and Petter Ögren. Enhancing UGV Teleoperation Control and Interface with FLC and Network Connectivity Perception . In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2015.

[18] Roberto Calandra, Nakul Gopalan, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. *Proceedings of the 8th International Conference on Learning and Intelligent Optimization*, chapter Bayesian Gait Optimization for Bipedal Locomotion, pages 274–290. Springer International Publishing, Cham, 2014. ISBN 978-3-319-09584-4. doi: 10.1007/978-3-319-09584-4_25.

[19] J. Carlson and Robin R. Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3):423–437, 2005.

[20] Jennifer Casper and Robin R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 33(3):367–385, 2003.

[21] F. Colas, S. Mahesh, F. Pomerleau, Ming Liu, and R. Siegwart. 3d path planning and execution for search and rescue ground robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 722–727, 2013.

[22] Joseph Sun Cruz, Dana Kulić, and William Owen. *Proceedings of the Second International Conference on Autonomous and Intelligent Systems*, chapter Online Incremental Learning of Inverse Dynamics Incorporating Prior Knowledge, pages 167–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-21538-4. doi: 10.1007/978-3-642-21538-4_17.

[23] Paulo G De Barros, Robert W Lindeman, and Matthew O Ward. Enhancing robot teleoperator situation awareness and performance using vibro-tactile and graphical feedback. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 47–54. IEEE, 2011.

[24] J.S. de la Cruz, W. Owen, and D. Kulic. Online learning of inverse dynamics via gaussian process regression. In *IROS*, pages 3583–3590, 2012. doi: 10.1109/IROS.2012.6385817.

[25] Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2 (12):1–142, 2011. doi: 10.1561/2300000021.

[26] Anna Derbakova, Nikolaus Correll, and Daniela Rus. Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3863–3868. IEEE, 2011.

[27] Gianni A Di Caro, Eduardo Feo Flushing, and Luca M Gambardella. Use of time-dependent spatial maps of communication quality for multi-robot path planning. In *International Conference on Ad-Hoc Networks and Wireless*, pages 217–231. Springer, 2014.

[28] R. Diankov and J. Kuffner. Randomized statistical path planning. In *IROS*, pages 1–6, 2007. doi: 10.1109/IROS.2007.4399557.

[29] DJI. Dji phantom 3 pro. URL `http://www.dji.com/de/product/phantom-3-pro`. `http://www.dji.com/de/product/phantom-3-pro` [Online: 05.01.2016].

[30] Paulo Drews, Pedro Núñez, Rui P Rocha, Mario Campos, and Jorge Dias. Novelty detection and segmentation based on gaussian mixture models: A case study in 3d robotic laser mapping. *Robotics and Autonomous Systems*, 61(12):1696–1709, 2013.

[31] S. Duane, A. D. Kennedy, B. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195:216–222, 1987. doi: 10.1016/0370-2693(87)91197-X.

[32] Renaud Dubé, Abel Gawel, Luigi Freda, Mario Gianni, Cesar Cadena, and Roland Siegwart. 3d localization, mapping and path planning for search and rescue operations. In *Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2016.

[33] Mica R Endsley et al. Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, pages 3–32, 2000.

[34] F. Ferri, M. Gianni, M. Menna, and F. Pirri. Point cloud segmentation and 3d path planning for tracked vehicles in cluttered and dynamic environments. In *In Proceedings of the 3rd IROS Workshop on Robots in Clutter: Perception and Interaction in Clutter*, 2014.

[35] F. Ferri, M. Gianni, M. Menna, and F. Pirri. Point cloud segmentation and 3d path planning for tracked vehicles in cluttered and dynamic environments. In *In Proceedings of the 3rd IROS Workshop on Robots in Clutter: Perception and Interaction in Clutter*, 2014.

[36] F. Ferri, M. Gianni, M. Menna, and F. Pirri. Dynamic obstacles detection and 3d map updating. In *IROS*, pages 5694–5699, 2015. doi: 10.1109/IROS.2015.7354185.

[37] Federico Ferri, Mario Gianni, Matteo Menna, and Fiora Pirri. Point cloud segmentation and 3d path planning for tracked vehicles in cluttered and dynamic environments. In *Proc. of the 3rd IROS Workshop on Robots in Clutter: Perception and Interaction in Clutter*, 2014.

[38] Federico Ferri, Mario Gianni, Matteo Menna, and Fiora Pirri. Dynamic obstacles detection and 3d map updating. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5694–5699. IEEE, 2015.

[39] A. Fink, H. Beikirch, M. Voss, and C. Schroder. RSSI-based indoor positioning using diversity and Inertial Navigation. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2010.

[40] Terrence Fong and Charles Thorpe. Vehicle teleoperation interfaces. *Autonomous robots*, 11(1):9–18, 2001.

[41] M Frejek and Scott B Nokleby. A methodology for tele-operating mobile manipulators with an emphasis on operator ease of use. *Robotica*, 31(03):331–344, 2013.

[42] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[43] Albert Clerigues Garcia. Traversing algorithm based on proprioceptive measures for sensory deprived environments. Master's thesis, Czech Technical University in Prague, Czech Republic, 2016.

[44] Aa Ghaffarkhah and Ya Mostofi. Channel learning and communication-aware motion planning in mobile networks. In *American Control Conference (ACC), 2010*, pages 5413–5420, June 2010.

[45] M. Gianni, M. Garcia, F. Ferri, and F. Pirri. Terrain contact modeling and classification for atvs. In *ICRA*, pages 186–192, 2016. doi: 10.1109/ICRA.2016.7487132.

[46] Mario Gianni, Gianpaolo Gonnelli, Arnab Sinha, Matteo Menna, and Fiora Pirri. An augmented reality approach for trajectory planning and control of tracked vehicles in rescue environments. In *Proceedings of the 11th IEEE International Symposium on Safety, Security and Rescue Robotics, Linkoping, Sweden*, 2013.

[47] Mario Gianni, Federico Ferri, Matteo Menna, and Fiora Pirri. Adaptive robust three-dimensional trajectory tracking for actively articulated tracked vehicles. *Journal of Field Robotics*, pages n/a–n/a, 2015. ISSN 1556-4967. doi: 10.1002/rob.21584. URL `http://dx.doi.org/10.1002/rob.21584`.

[48] Mario Gianni, Manuel A. Ruiz Garcia, and Fiora Pirri. Learning the dynamics of articulated tracked vehicles. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(6):155 – 158, 2016. ISSN 1307-6892. URL `http://iastem.com/Publications?p=114`.

[49] Arjan Gijsberts and Giorgio Metta. Real-time model learning using incremental sparse spectrum gaussian process regression. *Neural Networks*, 41:59–69, 2013. doi: http://dx.doi.org/10.1016/j.neunet.2012.08.011.

[50] D Girardeau-Montaut, M Roux, R Marc, and G Thibault. Change detection on points cloud data acquired with a ground laser scanner. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(part 3):W19, 2005.

[51] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25 (7):53–82, 2010. ISSN 0885-8985. doi: 10.1109/MAES.2010.5546308.

[52] Andreas Hedström, Henrik I Christensen, and Carl Lundberg. A wearable gui for field robots. In *Field and Service Robotics*, pages 367–376. Springer, 2006.

[53] Susan Hert, Sanjay Tiwari, and Vladimir Lumelsky. *A Terrain-Covering Algorithm for an AUV*, pages 17–45. Springer US, Boston, MA, 1996. ISBN 978-1-4613-1419-6. doi: 10.1007/978-1-4613-1419-6_2. URL http://dx.doi.org/10.1007/978-1-4613-1419-6_2.

[54] D. Hestand and H.A. Yanco. Layered sensor modalities for improved human-robot interaction. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2966–2970 vol.3, Oct 2004. doi: 10.1109/ICSMC.2004.1400784.

[55] Hokuyo. Hokuyo. URL http://www.hokuyo-aut.jp/. http://www.hokuyo-aut.jp/ [Online: 20.01.2016].

[56] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[57] M Ani Hsieh, Anthony Cowley, Vijay Kumar, and Camillo J Taylor. Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1-2):111–131, 2008.

[58] Tor Arne Johansen et al. Task assignment for cooperating uavs under radio propagation path loss constraints. In *2012 American Control Conference (ACC)*, pages 3278–3283. IEEE, 2012.

[59] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy. A bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4):383–400, 2011. ISSN 0929-5593. doi: 10.1007/s10514-011-9248-x. URL http://dx.doi.org/10.1007/s10514-011-9248-x.

[60] K. Knipp. "Selbstlokalisation eines Mikroflugsystems mit Laserscannern zur 3D-Kartierung von Innenräumen", Bachelor's thesis, HBRS University of Applied Sciences, March 2016. URL http://publica.fraunhofer.de/. http://publica.fraunhofer.de/dokumente/.

[61] Oussama Kanoun, Florent Lamiraux, Pierre-Brice Wieber, Fumio Kanehiro, Eiichi Yoshida, and Jean-Paul Laumond. Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2939–2944. IEEE, 2009.

[62] Christos Katsikiotis, Dimitrios Zorbas, and Periklis Chatzimisios. *Connectivity Restoration and Amelioration in Wireless Ad-Hoc Networks: A Practical Solution*, pages 255–264. Springer International Publishing, Cham, 2014. ISBN 978-3-319-13329-4. doi: 10.1007/978-3-319-13329-4_23.

[63] Kyu-Han Kim, Kang G Shin, and Dragos Niculescu. Mobile autonomous router system for dynamic (re) formation of wireless relay networks. *IEEE Transactions on Mobile Computing*, 12(9):1828–1841, 2013.

[64] Jens Kober, Andreas Wilhelm, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, 33(4):361–379, 2012. doi: 10.1007/s10514-012-9290-3.

[65] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160, 2011.

[66] O.B. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *Robotics and Autonomous Systems*, 58(9):1105–1116, 2010. doi: http://dx.doi.org/10.1016/j.robot.2010.06.001.

[67] Philipp Krusi, Paul Furgale, Michael Bosse, and Roland Siegwart. Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments. *Journal of Field Robotics*, pages n/a–n/a, 2016. ISSN 1556-4967. doi: 10.1002/rob.21700. URL `http://dx.doi.org/10.1002/rob.21700`.

[68] B Larochelle, Geert-Jan Kruijff, N Smets, T Mioch, and P Groenewegen. Establishing human situation awareness using a multi-modal operator control unit in an urban search & rescue human-robot team. *RO-MAN, 2011 IEEE*, pages 229–234, 2011.

[69] Benoit Larochelle and GM Kruijff. Multi-view operator control unit to improve situation awareness in usar missions. In *RO-MAN, 2012 IEEE*, pages 1103–1108. IEEE, 2012.

[70] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Iowa State Univerisity, 1998.

[71] Daniel J. Lizotte, Tao Wang, Michael H. Bowling, and Dale Schuurmans. Automatic gait optimization with gaussian process regression. In *IJCAI*, 2007.

[72] Ma Malmirchegini and Ya Mostofi. On the spatial predictability of communication channels. *IEEE Transactions on Wireless Communications*, 11(3):964–978, March 2012.

[73] Matteo Menna, Mario Gianni, Federico Ferri, and Fiora Pirri. Real-time autonomous 3d navigation for tracked vehicles in rescue environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.

[74] Byung-Cheol Min, Yongho Kim, Sangjun Lee, Jin-Woo Jung, and Eric T. Matson. Finding the optimal location and allocation of relay robots for building a rapid end-to-end wireless communication. *Ad Hoc Networks*, 39:23 – 44, 2016. ISSN 1570-8705. doi: http://dx.doi.org/10.1016/j.adhoc.2015.12.001.

[75] Benoit Morisset, Radu Bogdan Rusu, Aravind Sundaresan, Kris Hauser, Motilal Agrawal, Jean-Claude Latombe, and Michael Beetz. Leaving flatland: Toward real-time 3d navigation. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3786–3793, 2009.

[76] Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Daniel M. Helmick, and Larry Matthies. Autonomous stair climbing for tracked vehicles. *Int. J. Rob. Res.*, 26(7):737–758, 2007.

[77] Robin R Murphy. *Disaster Robotics*. MIT Press, 2014.

[78] R.R. Murphy. Human–Robot Interaction in Rescue Robotics. *IEEE Transactions on systems, Man, and Cybernetics, Part C: Application and Reviews*, 34(2), 2004.

[79] Sebastian Muszynski, Jorg Stuckler, and Sven Behnke. Adjustable autonomy for mobile teleoperation of personal service robots. *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 933–940, September 2012. doi: 10.1109/ROMAN.2012.6343870.

[80] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and Shinji Kawatsuma. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013. ISSN 1556-4967. doi: 10.1002/rob.21439.

[81] Ramviyas Nattanmai Parasuraman, Alessandro Masi, and Manuel Ferre. *Wireless Communication Enhancement Methods for Mobile Robots in Radiation Environments. Radio communication for robotic application at CERN.* PhD thesis, Madrid, Polytechnic U., Oct 2014. URL `https://cds.cern.ch/record/1956445`. Presented 17 Oct 2014.

[82] Duy Nguyen-Tuong, B. Scholkopf, and J. Peters. Sparse online model learning for robot control with support vector regression. In *IROS*, pages 3121–3126, 2009. doi: 10.1109/IROS.2009.5354609.

[83] Petter Ögren, Peter Svenmarck, Patrik Lif, Martin Norberg, and Nils Emil Söderbäck. Design and Implementation of a New Teleoperation Control Mode for Differential Drive UGVs. *Autonomous Robots*, 2014.

[84] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In *ICRA*, pages 4029–4036, 2014. doi: 10.1109/ICRA.2014.6907444.

[85] Ramviyas Parasuraman, Thomas Fabry, Luca Molinari, Keith Kershaw, Mario Di Castro, Alessandro Masi, and Manuel Ferre. A Multi-Sensor RSS Spatial Sensing-Based Robust Stochastic Optimization Algorithm for Enhanced Wireless Tethering. *Sensors*, 14(12):23970–24003, 2014. ISSN 1424-8220. doi: 10.3390/s141223970.

[86] Parrot. Parrot bebop drone. URL `http://www.parrot.com/de/produkte/bebop-drone/`. `http://www.parrot.com/de/produkte/bebop-drone/` [Online: 05.01.2016].

[87] Ira Pohl. A method for finding hamilton paths and knight's tours. *Communications of the ACM*, 10(7):446–449, 1967.

[88] F. Pomerleau, P. Krsi, F. Colas, P. Furgale, and R. Siegwart. Long-term 3d map maintenance in dynamic environments. In *ICRA*, pages 3712–3719, 2014. doi: 10.1109/ICRA.2014.6907397.

[89] François Pomerleau, Philipp Krüsi, Francis Colas, Paul Furgale, and Roland Siegwart. Long-term 3d map maintenance in dynamic environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3712–3719. IEEE, 2014.

[90] M. Rank, Z. Shi, H. J. Mller, and S. Hirche. Predictive communication quality control in haptic teleoperation with time delay and packet loss. *IEEE Transactions on Human-Machine Systems*, 46(4):581–592, Aug 2016. ISSN 2168-2291. doi: 10.1109/THMS.2016.2519608.

[91] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *NIPS*, pages 881–888. MIT Press, 2002.

[92] Justin Richer and Jill L Drury. A video game-based framework for analyzing human-robot interaction: characterizing interface design in real-time interactive multimedia applications. In *Proceedings of the*

1st ACM SIGCHI/SIGART conference on human-robot interaction, pages 266–273. ACM, 2006.

[93] Carlos Rizzo, Domenico Sicignano, L. Riazuelo, D. Tardioli, Francisco Lera, José Luis Villarroel, and L. Montano. *Guaranteeing Communication for Robotic Intervention in Long Tunnel Scenarios*, pages 691–703. Springer International Publishing, Cham, 2016. ISBN 978-3-319-27146-0. doi: 10.1007/978-3-319-27146-0_53.

[94] Louis B Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 76–82. IEEE, 1993.

[95] M. Schwarz and S. Behnke. Local navigation in rough terrain using omnidirectional height. In *Proocedings of the International Symposium on Robotics*, pages 1–6, June 2014.

[96] Vikas Shivashankar, Rajiv Jain, Ugur Kuter, and Dana S Nau. Real-time planning for covering an initially-unknown spatial environment. In *FLAIRS Conference*, 2011.

[97] Nanja JJM Smets, Guido M te Brake, Mark A Neerincx, and Jasper Lindenberg. Effects of mobile map orientation and tactile feedback on navigation speed and situation awareness. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 73–80. ACM, 2008.

[98] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *NIPS*, 2012.

[99] Cyrill Stachniss, Udo Frese, and Giorgio Grisetti. Openslam. URL `https://openslam.org/`. `https://openslam.org/` [Online: 27.11.2016].

[100] Maciej Stachura and Eric W Frew. Cooperative target localization with a communication-aware unmanned aircraft system. *Journal of Guidance, Control, and Dynamics*, 34(5):1352–1362, 2011.

[101] Thuy T. Truong, Kenneth N. Brown, and Cormac J. Sreenan. An online approach for wireless network repair in partially-known environments. *Ad Hoc Networks*, 45:47 – 64, 2016. ISSN 1570-8705. doi: http://dx.doi.org/10.1016/j.adhoc.2016.02.021.

[102] Antonio W Vieira, Paulo LJ Drews, and Mario FM Campos. Efficient change detection in 3d environment for autonomous surveillance robots based on implicit volume. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2999–3004. IEEE, 2012.

[103] Sethu Vijayakumar and Stefan Schaal. Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space. In *ICML*, 2000.

[104] V. Šalanský, V. Kubelka, K. Zimmermann, M. Reinstein, and T. Svoboda. Touching without vision: terrain perception in sensory deprived environments. In *21st Computer Vision Winter Worskhop*, 2016.

[105] R. Wegner and J. Anderson. Agent-Based Support for Balancing Teleoperation and Autonomy in Urban Search and Rescue. *International Journal of Robotics and Automation*, 21(2):1–19, 2006. ISSN 1925-7090. doi: 10.2316/Journal.206.2006.2.206-2796.

[106] W. Wei, H.and Lu, P. Zhu, S. Ferrari, R. H. Klein, S. Omidshafiei, and J. P. How. Camera control for learning nonlinear target dynamics via bayesian nonparametric dirichlet-process gaussian-process (dp-gp) models. In *IROS*, pages 95–102, 2014.

[107] Christopher D Wickens. Multiple resources and mental workload. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):449–455, 2008.

[108] Fang-Jing Wu, Yu-Fen Kao, and Yu-Chee Tseng. From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile Computing*, 7(4):397–413, 2011.

[109] Brian M Yamauchi. Packbot: a versatile platform for military robotics. In *Defense and Security*, pages 228–237. International Society for Optics and Photonics, 2004.

[110] Yuan Yan and Yasamin Mostofi. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *IEEE Transactions on Wireless Communications*, 12(4):1562–1572, 2013.

[111] H.A. Yanco and J. Drury. Where Am I? Acquiring Situation Awareness Using a Remote Robot Platform. *IEEE Conference on Systems, Man and Cybernetics*, 2004.

[112] Holly A Yanco and Jill L Drury. Rescuing interfaces: A multi-year study of human-robot interaction at the aaai robot rescue competition. *Autonomous Robots*, 22(4):333–352, 2007.

[113] Holly A Yanco, Michael Baker, Robert Casey, Brenden Keyes, Philip Thoren, Jill L Drury, Douglas Few, Curtis Nielsen, and David Bruemmer. Analysis of human-robot interaction for urban search and rescue. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 22–24, 2006.

[114] Tomoaki Yoshida, Keiji Nagatani, Satoshi Tadokoro, Takeshi Nishimura, and Eiji Koyanagi. Improvements to the rescue robot quince toward future indoor surveillance missions in the fukushima daiichi nuclear power plant. In *Field and Service Robotics*, pages 19–32. Springer, 2014.

[115] Bo Zhang, Yunlong Wu, Xiaodong Yi, and Xuejun Yang. Joint communication-motion planning in wireless-connected robotic networks: Overview and design guidelines. *arXiv preprint arXiv:1511.02299*, 2015.

2 Annexes

2.1 Baberg (2016), "Adaptive Object Centered Teleoperation Control of a Mobile Manipulator"

Bibliography Baberg. "Adaptive Object Centered Teleoperation Control of a Mobile Manipulator" In *Proceedings of the IEEE ICRA 2016*, Stockholm, Sweden, 2016.

Abstract Teleoperation of a mobile robot manipulating and exploring an object shares many similarities with the manipulation of virtual objects in a 3D design software such as AutoCAD. The user interfaces are however quite different, mainly for historical reasons. In this paper we aim to change that, and draw inspiration from the 3D design community to propose a teleoperation interface control mode that is identical to the ones being used to locally navigate the virtual viewpoint of most Computer Aided Design (CAD) softwares.

The proposed mobile manipulator control framework thus allows the user to focus on the 3D objects being manipulated, using control modes such as *orbit object* and *pan object*, supported by data from the wrist mounted RGB-D sensor. The gripper of the robot performs the desired motions relative to the object, while the manipulator arm and base moves in a way that realizes the desired gripper motions. The system redundancies are exploited in order to take additional constraints, such as obstacle avoidance, into account, using a constraint based programming framework.

Relation to WP This paper describes a mobile manipulation approach used in T2.3.

Availability Unrestricted.

2.2 Parasuraman (2016), "A New UGV Teleoperation Interface for Improved Awareness of Network Connectivity and Physical Surroundings"

Bibliography Ramviyas Parasuraman, Sergio Caccamo, Fredrik Båberg, Mark Neerincx, Petter Ögren. "A New UGV Teleoperation Interface for Improved Awareness of Network Connectivity and Physical Surroundings." Unpublished report, December 2016.

Abstract Recent and current Urban Search and Rescue (USAR) missions show that the range and coverage of the wireless connection between the operator and the teleoperated Unmanned Ground Vehicle (UGV) presents

a significant constraint on the mission execution. For continuing operation, the operator needs to continuously adapt to the dynamic network connectivity across the environment in addition to performing the primary navigation, observation and manipulation tasks.

In this paper, a new teleoperation User Interface (UI) is presented that integrates information on the Direction of Arrival (DoA) of the radio signal. The proposed approach consists of (1) a method for estimating the DoA and (2) a color-bar representation surrounding the video feed that informs the operator which navigation directions of motion are safe, even when moving in regions close to the connectivity threshold.

The UI was evaluated in a user study with 24 participants who performed a search task under challenging wireless connectivity conditions. The results show that using the proposed interface resulted in more objects found, and less missions aborted due to connectivity problems, as compared to a standard interface.

Relation to WP This paper describes network aware UGV teleoperation. T2.3

Availability Restricted. Unpublished report.

2.3 Knipp (2016), "Self localisation of a micro aerial vehicle with laser scanners for indoor 3D-mapping (bachelor's thesis)"

Bibliography Kalle Knipp. "Selbstlokalisation eines Mikroflugsystems mit Laserscannern zur 3D-Kartierung von Innenräumen." Bachelor's thesis, HBRS University of Applied Sciences Sankt Augustin, March 2016.

Abstract The bachelor's thesis "self localisation of a micro aerial vehicle with laser scanners for indoor 3D-mapping" presents a procedure to determine the position of a micro arial vehicle depending on all of its six dimensions of freedom. A 3D map is created simultaneously by using two laser scanners. Hector SLAM is used to determine the movement of the arial vehicle in the horizontal plain. To gain additional altitude information two different techniques are implemented: The first one measures altitude information directly based on the data of one laser scanner, the second one uses Hector SLAM to obtain altitude information from a vertical scan plain.

Relation to WP Within this thesis, a method for UAV localization has been developed, to be used in the context of T2.6.

Availability Unrestricted. Available at `www.urlhere.com`.

2.4 Garcia-bc-thesis-2016, "Traversing algorithm based on proprioceptive measures for sensory deprived environments (bachelor's thesis)"

Bibliography Garcia, Albert Clerigues. Traversing algorithm based on proprioceptive measures for sensory deprived environments. Bachelor Thesis, Czech Technical Univeristy in Prague, 2016.

Abstract The main goal of the diploma project is to develop a combined robot-motion and tilting-flippers algorithm for traversing harsh terrains without extero- ceptive measurements, i.e. without Lidar, RGB-D cameras, etc. The robot should advance in the direction given by the operator as autonomously as possible, deciding how much distance to advance, positioning the flippers to aid in traversal and ensuring its safety. The features considered in the planning will be exclusively proprioceptive such as tactile sensor arrays in each flipper, gyroscopes, accelerometers or odometry.

Relation to WP The thesis propose a method for autonomous UGV motion without using exteroceptive sensing (no cameras, no lidar) T2.6.

Availability Unrestricted. Available at `ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/Garcia-bc-thesis-2016.pdf`

2.5 Burian-ms-thesis-2017, "Haptic Terrain Exploration with Robotic Arm"

Bibliography Burian, Martin. Haptic Terrain Exploration with Robotic Arm. Master Thesis, Czech Technical University in Prague, 01/2017.

Abstract The UGV sensing can fail due to environmental conditions like smoke and dust, making further robot operation difficult and dangerous for the robot. In such cases, the robot-mounted robotic arm can be used to gain information about the environment by obtaining tactile measurements from it. We propose an algorithm to guide the arm during the exploration, along with methods to control the arm movement and to gather the measurements with a 3D force sensor from safe distance. The system is implemented and put to the test in both simulated experiments and real world trials.

Relation to WP Arm based tactile terrain exploration. T2.6.

Availablity Unrestricted. Available at `ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/Burian-ms-thesis-2017.pdf`

2.6 Dube (2016), "3D Localization, Mapping and Path Planning for Search and Rescue Operations"

Bibliography R. Dube, A. Gawel, C. Cadena, R. Siegwart, L. Freda, M. Gianni. "3D Localization, Mapping and Path Planning for Search and Rescue Operations." In *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Lausanne, Switzerland, 2016.

Abstract This work presents our results on 3D robot localization, mapping and path planning for the latest joint exercise of the European project Long-Term Human-Robot Teaming for Robots Assisted Disaster Response (TRADR). The full system is operated and evaluated by firemen end-users in real-world search and rescue experiments. We demonstrate that the system is able to plan a path to a goal position desired by the fireman operator in the TRADR Operational Control Unit (OCU), using a persistent 3D map created by the robot during previous sorties.

Relation to WP This work contributes to Task T2.6 in reporting the efforts done in integrating the path planning framework developed by ROMA with the mapping pipeline of ETHZ. Several new features have also been integrated and tested into the framework. Results are reported in the description of the exercises.

Availability Unrestricted. Included in the public version of this deliverable.

2.7 Caccamo (2017), "Autonomous repair of wireless network in mobile robots with resilient communication-aware motion planner"

Bibliography Sergio Caccamo, Ramviyas Parasuraman, Luigi Freda, Mario Gianni, Petter Ogren. "Autonomous repair of wireless network in mobile robots with resilient communication-aware motion planner." Unpublished report, December 2016.

Abstract In this paper, we present a robust and online radio signal mapping method using Gaussian Random Field, and propose a Resilient Communication Aware Motion Planner (RCAMP) that integrates the above signal mapping framework with a motion planner that considers environment and physical constraints of the robot based on the available sensory information. We also propose a self-repair strategy using the proposed RCMAP to drive the robot to the connection-safe position prioritizing to reach a destination (if provided), in the event of a communication loss. We demonstrate

the proposed planner with realistic simulations in different scenarios of an exploration task in single or multi-channel communication.

Relation to WP This work contributes to Task T2.6 in proposing a preliminary approach for signal coverage mapping and path planning under time-space signal state constraints. The objective of this work is to provide a robust solution to unexpected wireless connection drops. This research has been performed in collaboration between KTH and Roma. Due to inaccuracies of the signal coverage mapping model, further research work is still required to consider this functionality effective and reliable.

Availability Unrestricted. Included in the public version of this deliverable.

2.8 Freda (2017), "An hybrid approach for trajectory control design"

Bibliography Luigi Freda, Mario Gianni, Fiora Pirri. "An hybrid approach for trajectory control design." Unpublished report, December 2016.

Abstract This work presents a methodology to design trajectory tracking feedback control laws, which embed non-parametric statistical models, such as Gaussian Processes (GPs). The aim is to minimize unmodeled dynamics such as undesired slippages. The proposed approach has the benefit of avoiding complex terramechanics analysis to directly estimate from data the robot dynamics on a wide class of trajectories. Experiments in both real and simulated environments prove that the proposed methodology is promising.

Relation to WP This work contributes to Task T2.6 in developing a framework for estimating a GP-based second-order inverse dynamic model of the TRADR UGV. The model integrated into the feedback control loop demonstrated good performance.

Availability Restricted. Included in the public version of this deliverable.

Adaptive Object Centered Teleoperation Control of a Mobile Manipulator

Fredrik Båberg, Yuquan Wang, Sergio Caccamo, Petter Ögren

***Abstract*— Teleoperation of a mobile robot manipulating and exploring an object shares many similarities with the manipulation of virtual objects in a 3D design software such as AutoCAD. The user interfaces are however quite different, mainly for historical reasons. In this paper we aim to change that, and draw inspiration from the 3D design community to propose a teleoperation interface control mode that is identical to the ones being used to locally navigate the virtual viewpoint of most Computer Aided Design (CAD) softwares.**

The proposed mobile manipulator control framework thus allows the user to focus on the 3D objects being manipulated, using control modes such as *orbit object* and *pan object*, supported by data from the wrist mounted RGB-D sensor. The gripper of the robot performs the desired motions relative to the object, while the manipulator arm and base moves in a way that realizes the desired gripper motions. The system redundancies are exploited in order to take additional constraints, such as obstacle avoidance, into account, using a constraint based programming framework.

***Index Terms*— Virtual object, mobile manipulation, teleoperation**

I. Introduction

Teleoperated mobile robots equipped with manipulators are expected to play key roles in future *Search and Rescue* and *Explosive Ordnance Disposal* operations. In these applications, robots are sent to places where it is not safe for humans to go, but difficult tasks still have to be carried out.

It is well known that robot teleoperation is a demanding task [1], and a lot of research is currently aimed to improve performance and reduce operator workload in these safety critical applications.

It has been noted that robot teleoperation has many similarities with playing first person perspective computer games [2]. In both cases a human is controlling an entity that is moving around in a remote environment trying to achieve a specific task. These similarities have been used to improve many parts of the teleoperation, from using gamepads for input, to designing control modes and the presentation of video streams and other sensing modalities.

In this paper, we draw inspiration from another area of virtual reality. Instead of computer games, we look at the interfaces of *3D design software* such as Autodesk AutoCAD, SolidWorks, V-REP[1] and Gazebo[1]. These software tools are used by engineers and architects to make 3D drawings and designs. When manipulating objects, the users navigate the virtual space using the functions *pan object* and *orbit object*, see Figure 1. These functions are the core part of an interface that is used daily by thousands of professionals and has been refined in many iterations. Therefore, there is reason to believe that the same functions would be useful when exploring and manipulating remote objects with a teleoperated mobile manipulator equipped with a RGB-D sensor[2].

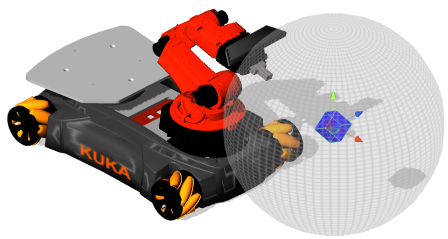


Fig. 1. Concept illustration. A mobile manipulator with a virtual sphere. The blue cube in the center of the sphere is the object to be examined. In the control mode *orbit object*, the end effector moves on the sphere while keeping the object in the center of view, when commanded left/right and up/down.

With the proposed approach, the user can choose between *pan object* and *orbit object* when controlling the robot. In both control modes, the full pose of the end effector, position and orientation is controlled using a gamepad, and the video stream from the RGB-D sensor mounted on the wrist of the end effector is shown to the user.

In *pan object*, the sensor equipped end effector moves in a so-called robot centric way, known from the literature, see e.g. [3]. A requested translation forwards results in the end effector moving on a straight line in the direction towards whatever is in the center of view, and a requested translation to the right results in a straight line to the right.

In *orbit object*, the motions are *object centric*, and relative to the object in the center of the sensor view. Also here, as for *pan object*, a requested forward translation results in a

The authors are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology (KTH), SE-100 44 Stockholm, Sweden. e-mail: {fbaberg|yuquan|caccamo|petter}@kth.se

[1]The last two examples are actually robot simulators, but this functionality concerns creating 3D environments

[2]Red Green Blue plus Depth, image including depth/distance information. For instance Microsoft Kinect, Intel RealSense, PrimeSense.

straight line motion towards the object in the center of the view. A requested translation to the right on the other hand, results in a circular arc trajectory orbiting the object in center of the view, and keeping that object in the center of the view by a corresponding rotation, see Figure 1. The radius of the orbit motion is given by the current distance to the center of the object, which is estimated using the RGB-D sensor.

The two control modes described above complement each other, and are often used in an alternating fashion. However, as *pan object* is equivalent with robot centric control, [3], we focus this work on *orbit object* which is new to the robotics community.

In the design applications, *orbit object* is useful for exploring an object, and moving into a viewpoint that allow you to add or remove details. In the robot teleoperation application, we believe that *orbit object* would be useful for exploring objects, getting good camera views from all sides. It would also be useful when gathering 3D data from the RGB-D sensor, in order to create a high quality 3D-model. Finally, it would be convenient when deciding on the appropriate grasp point for lifting an object, or operating a door handle.

The approach presented in this paper realizes the *orbit object* control mode using constraint based programming. To realize the appropriate motion of the end effector, the configuration of the complete mobile manipulator must be taken into account. Sometimes it is best to move only the arm, but sometimes the mobile base has to be moved to increase the range of the arm, while simultaneously taking obstacles and internal singularities into account. All this has to happen automatically, enabling the user to focus on the task at hand, which is being carried out by moving the end effector relative to the object of interest.

The contribution of this paper is that we show how to realize the control mode *orbit object* in a teleoperated mobile manipulator. To the best of our knowledge, this has not been done before. We also show how to incorporate avoidance of obstacles into the framework using constraint based programming.

The structure of the paper is as follows. First, Section II describes related work. Then, in Section III we will provide some notation and definitions, and formulate the problem, before proposing a solution in Section IV. The solution is verified with experiments in Section V, and finally conclusions can be found in Section VI.

II. Related Work

As the proposed approach involves teleoperation of a mobile manipulator, we will first discuss work on teleoperation of mobile robots, and then manipulators.

Within the area of search and rescue robotics there has been a lot of work on teleoperation of mobile robots, and a nice overview of the problems involved can be found in [1] and [4]. While [1] describes the domain in detail, [4] suggest possible improvements in terms of multimodal feedback, such as using combinations of video, audio, and haptics.

In a study based on experiences from the AAAI Robot Rescue Competitions in 2002-2004 [5], the authors noticed an evolution over time, towards a large single interface, with a large percentage of the screen dedicated to video.

The idea of supporting user situation awareness with a virtual 3D rendering of the robot and its surroundings was explored in [6] and [7] and the use of multi-touch Operator Control Units (OCUs) including fusion of sensor information to lower the operator's cognitive load was investigated in [8].

In [9] the authors identify seven fundamental problems in OCU design, propose a solution focussing on sensor data presentation, and present results from end-user evaluations.

The proposed paper differs from the work above in that none of the above consider the actual control layer of the OCU, instead they focus on how information is presented to the operator.

Within the area of mobile robot control, a lot of inspiration has been drawn from similarities between computer games, and robot teleoperation, and [2] is an early study on this topic. There it is argued that Video Game Based Frameworks (VGBF) are very useful for both evaluating existing interfaces and inspiring the design of new ones. The authors then go on to make a detailed categorization of input and output devises as well as methods used in different games and discuss different combinations of real video streams and rendered images of the vehicle surroundings.

One way of using inspiration from computer games was presented in [10], where the classical robot control mode of *Tank Control* was replaced with *Free Look Control* which is used in many computer games.

In our work, we draw inspiration from virtual interfaces, but our inspiration comes not from computer games, but from professional modeling tools such as AutoCAD.

There has also been a lot of studies into the area of teleoperation for manipulation. The importance of different reference frames, i.e. robot centric or view centric, was explored in [3], where the author propose an Ecological interface design that aims to make relationships in the environment perceptually evident to the user, in order to minimize the effort needed for understanding those relationships.

The use of smartphones or tablets to control a manipulator was investigated in [11], where the operator could either modify the target position of the end effector in the workspace, or use the high level skill of autonomous grasping.

The effect of stereoscopic displays on task performance and cognitive workload was investigated in [12], and performance on different autonomy levels was studied in [13]. The levels included direct control, waypoint control, indication of general grasps area, and completely autonomous grasping.

Performing manipulation with user input in terms of 2D click and drag input from a mouse was explored in [14]. There, five different strategies were investigated, including joint space control, cartesian space control, and 3 versions of obstacle avoidance based on reactive control, filtered prediction and motion planning.

Teleoperation of a 8 DoF mobile manipulator using a 6

DoF joystick was investigated in [15]. The authors propose a control approach where the user controls the gripper pose, while the mobile base adapts and follows the gripper when possible and needed, to avoid over extending the arm.

The approach proposed in this paper differs from all the above in that the control mode is neither robot centric nor world centric, but object centric. In the *orbit object* control mode, a commanded motion to the right results in moving right with respect to the gripper, but keeping a constant distance to the object. This is motivated by an argument similar to the ones suggesting inspiration from computer game interfaces [2], but this time the inspiration comes from the professional 3D design community.

Motions can be constrained through virtual fixtures [16], and the approach used in this paper can be seen as such in the sense that regions can be restricted through constraints. However our implementation does not use force sensors for feedback, which is the case in for instance [17].

Introducing different constraints for restricting motion could introduce conflicts, in which case it could be necessary to prioritize [18]. In this paper we do not explicitly consider priorities, however by changing weights and introducing slack variables this could be considered.

III. Problem Formulation

In this section, we describe *orbit object* in more detail, and establish the notation used in the paper.

Boldface will indicate vectors, and the indices w, b, a, e, o denote **w**orld, robot **b**ase, **a**rm base, **e**nd effector and **o**bject respectively. The frames can be seen in Figures 2 and 3.

$\mathbf{q}$ - joint positions
$\mathbf{p}^i_j$ - position of object j in frame i
$r(t)$ - distance between end effector and object
J - Spatial Jacobian
f_i - constraints
$Ad_{g_{xy}}$ - Adjoint transformation from x to y
$\mathbf{e}^i_j$ - unit vector j in frame i
$\dot{\mathbf{p}}_{ij}$ - velocity of object j with respect to object i (in frame i).
R^i_j - Rotation matrix of object j in relation to frame i.
$\mathbf{v}^e_d$ - desired end effector velocity (user input)
ω^e_d - desired end effector angular velocity (user input)
$\mathbf{u}$ - joint velocity for arm+base, from optimization problem.
I_e, I_{ie} - Set of indices for equalities and inequalities.

A dot above a symbol indicates differentiation with respect to time, e.g. $\dot{\mathbf{q}}$ denotes the joint velocities. Superscript indicates which frame is used. We now define the control mode.

Orbit Object To move on the surface of a sphere, centered on the object, with a fixed radius. In Figure 2 this would correspond to moving along the surface of the sphere, with constant radius $r(t)$, and with x-axis at $\{e\}$ aligned with the vector between $\{e\}$ and $\{o\}$, i.e. always facing the centre of the sphere.

Given the definition above, we aim to solve the following problem.

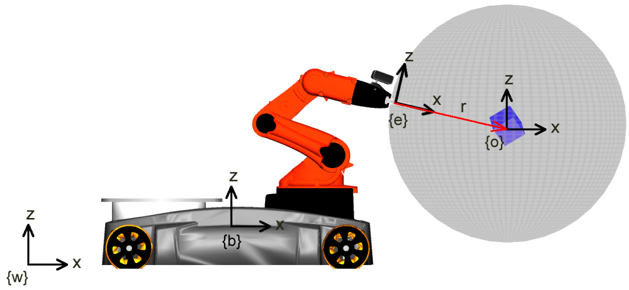


Fig. 2. Illustration of frames and points defined, side view.

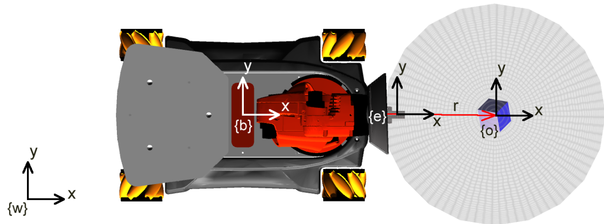


Fig. 3. Illustration of frames and points defined, top view.

Problem 3.1: Implement the control mode above, while avoiding collisions.

We will now describe the proposed solution.

IV. Proposed Solution

We propose to use a Constraint Based Programming (CBP) framework in order to solve Problem 3.1. Following the approach presented in [19] we first describe the problem we want to solve, and then state a reactive algorithm where a convex quadratic programming (QP) problem is solved in each timestep, taking the user input and the current state of all constraints into account.

Problem 4.1: Given a time interval $[t_0, t_f]$, initial state $q(t_0) = q_0$ and a control system

$$\dot{q} = h(q, u),$$

where $q \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Let us formulate the control objective in terms of a set of functions $f_i : \mathbb{R}^n \to \mathbb{R}$ and bounds $b_i \in \mathbb{R}$, $i \in I \subset \mathbb{N}$ as follows

$$\min_{u(\cdot)} \quad f_j(q(t_f), t_f), \; j \in I \tag{1}$$

$$\text{(s.t.)} \quad f_i(q(t), t) \leq b_i, \; \forall i \in I_{ie}, \; t > t_0 \tag{2}$$

$$f_i(q(t), t) = b_i, \; \forall i \in I_e, \; t > t_0 \tag{3}$$

where we assume that the constraints are satisfied at t_0, i.e. $f_i(q_0, t_0) \leq b_i$ for all $i \in I_{ie}$ and $f_i(q_0, t_0) = b_i$ for all $i \in I_e$ and $I_{ie}, I_e \subset I$.

Now, instead of addressing Problem 4.1 above directly, we look at a related problem where the constraints above are turned into feedback form, with controls moving the system back towards satisfying the constraints if they are momentarily not met due to e.g. uncertainties or disturbances. The related problem describes an online local controller, that also takes user input into account at each time step.

Problem 4.2:

$$\min_{u} \quad \dot{f}_j(q(t),u,t) + u^T Qu, \; j \in I \tag{4}$$

$$\text{(s.t.)} \quad \dot{f}_i(q,u,t) \le -k_i(f_i(q,t) - b_i), \; \forall i \in I_{ie}, \tag{5}$$

$$\dot{f}_i(q,u,t) = -k_i(f_i(q,t) - b_i), \; \forall i \in I_e, \tag{6}$$

where k_i are positive scalars and Q is a positive definite matrix.

First we look at the inequalities. It is clear that Equation (2) is satisfied for $t > t_0$ as long as Equation (5) is satisfied. Furthermore, in the worst case, if we have equality in Equation (5) then the bounds of Equation (2) will be exponentially approached, but not violated, with time constant $1/k_i$. Note that the bound will *only* be approached if motion in that direction corresponds to an improvement in the objective function, or is needed with respect to some other constraint.

Looking at the equalities, we also see that as long as Equations (6) are satisfied, so will (3), for $t > t_0$. Furthermore, if we have an error in the desired equality (3), then (6) will drive that error down to zero exponentially, with time constant $1/k_i$.

Then in the objective function, we know that (1) is kept small as long as its derivative $\dot{f}_j(q(t),u,t), j \in I$ is minimized. We smooth the input u by adding a quadratic regularization term $u^T Qu$ in (4), where Q is a diagonal positive-definite matrix designed to weight elements in u.

In order to address Problem 3.1 we need to provide a mapping between Problem 3.1 and 4.1. Then, we rely on the formalism above and iteratively solve Problem 4.2 in order to solve the two first ones.

Problem 3.1 can be captured in terms of the following constraints and corresponding equations.

- Keep desired distance from object, (7)
- Keep desired orientation w.r.t object, (8) and (9)
- Limit minimum end-effector altitude, (10)
- Avoid collision between robot and object, (11)
- Move according to user input. (12)

Formally, the constraints can be stated as

$$f_1 := ||\mathbf{p}_o^e||_2 = r_d \tag{7}$$

$$f_2 := \mathbf{p}_o^{e\top} \mathbf{e}_x^e - ||\mathbf{p}_o^e||_2 ||\mathbf{e}_x^e||_2 = 0, \tag{8}$$

$$f_3 := \mathbf{e}_z^{b\top} \mathbf{e}_y^e = 0, \tag{9}$$

$$f_4 := \mathbf{p}_e^{b\top} \mathbf{e}_z^b \ge z_{min}, \tag{10}$$

$$f_5 := ||\mathbf{p}_b^w - \mathbf{p}_{obs}^w||_2 \ge r_r, \tag{11}$$

$$f_6 := \dot{\mathbf{p}}_e^e = \mathbf{v}_d^e, \tag{12}$$

where r_d denotes desired distance between end-effector and object, given by the user. z_{min} is the minimum vertical separation of the end-effector and the robot base, r_r the minimum distance from obstacles and $\mathbf{v}_d^e$ is the desired end-effector movement given by user input. Only the movement in y- and z-direction, in the end-effector frame, is considered in constraint f_6 since the distance is given by constraint f_1.

Note that for readability there is a mixture of frames used in the constraints. Also note that there are inequalities in f_4, f_5 whereas the rest are equalities, thus $I_{ie} = \{4,5\}$ and $I_e = \{1,2,3,6\}$.

Having stated the constraints we now need to provide their time derivatives in order to formulate Problem 4.2. Details of how the derivatives were obtained can be found in the Appendix. In this paper we assume both the object to be inspected and the obstacle to be stationary. In the following, J_t and J_ω denotes the translational and rotational part of the Jacobian matrix. Unless otherwise stated, the Jacobian matrix is given in the world frame, $J = [Ad_{g_{wa}} J_{arm}, J_{base}]$.

$$\frac{\partial f_1}{\partial q} = -\frac{\mathbf{p}_o^{e\top} J_t}{\sqrt{\mathbf{p}_o^{e\top} \mathbf{p}_o^e}} \tag{13}$$

$$\frac{\partial f_2}{\partial q} = -\mathbf{p}_o^{e\top} S(R_e^w \mathbf{e}_x^e) J_\omega - (R_e^w \mathbf{e}_x^e)^\top J_t + \frac{\mathbf{p}_o^{e\top} J_t}{\sqrt{\mathbf{p}_o^{e\top} \mathbf{p}_o^e}} \tag{14}$$

$$\frac{\partial f_3}{\partial q} = -\mathbf{e}_z^{b\top} S(R_e^b \mathbf{e}_y^e) J_\omega^b \tag{15}$$

$$\frac{\partial f_4}{\partial q} = -\mathbf{e}_z^{b\top} J_t^b \tag{16}$$

$$\frac{\partial f_5}{\partial q} = -\frac{{\mathbf{p}_{obs}^b}^\top J_t}{\sqrt{{\mathbf{p}_{obs}^b}^\top \mathbf{p}_{obs}^b}} \tag{17}$$

$$\frac{\partial f_6}{\partial q} = J_t^e \tag{18}$$

Putting it all together we get Problem 4.3, the *orbit object* version of Problem 4.2. In this case we let $\dot{f}_j = 0$, as the youBot arm only has 5 DoFs.

Problem 4.3:

minimize $\quad \mathbf{u}(t)^\top \mathbf{Q} \cdot \mathbf{u}(t)$

subject to

$$\frac{\partial f_1}{\partial q}\mathbf{u}(t) = k_e(r_d - ||\mathbf{p}_o^w - \mathbf{p}_e^w||_2)$$

$$\frac{\partial f_2}{\partial q}\mathbf{u}(t) = k_e(0.0 - (\mathbf{p}_o^{e\top} R_e^w \mathbf{e}_x^e - ||\mathbf{p}_o^e||_2 ||R_e^w \mathbf{e}_x^e||_2))$$

$$\frac{\partial f_3}{\partial q}\mathbf{u}(t) = k_e(0.0 - (\mathbf{e}_z^{b\top} R_e^b \mathbf{e}_y^e))$$

$$\frac{\partial f_4}{\partial q}\mathbf{u}(t) \le -k_{ie}(z_d - (\mathbf{p}_e^{w\top} \mathbf{e}_z^w))$$

$$\frac{\partial f_5}{\partial q}\mathbf{u}(t) \le -k_{ie}(r_r - (||\mathbf{p}_b^w - \mathbf{p}_{obs}^w||_2))$$

$$\frac{\partial f_6}{\partial q}\mathbf{u}(t) = \mathbf{v}_d^e$$

where k_e, k_{ie} are weights for the equality- and inequality constraints.

V. Simulations

To illustrate the proposed approach, V-REP is used to simulate a KUKA youBot platform equipped with a youBot arm. On the sensor carrier of the arm, a kinect-like camera is mounted, providing RGB-D data.

The code is written in C++, and runs in Ubuntu 14.04 with ROS Indigo. The simulator has a scene with a youBot, equipped with an arm and a kinect camera, and an object to be examined. From V-REP the object location, expressed in the world frame, is obtained. Odometry data and joint states are provided as normal ROS topics. For repeatability, input is generated by given functions of time, but could easily be provided by user commands from a gamepad. Gurobi is used for solving the optimization problem.

The task is to examine a cube shaped object, with each side 0.2 m, using the *orbit object* control mode, as seen in Figure 4. This requires movement of both the arm and the base.

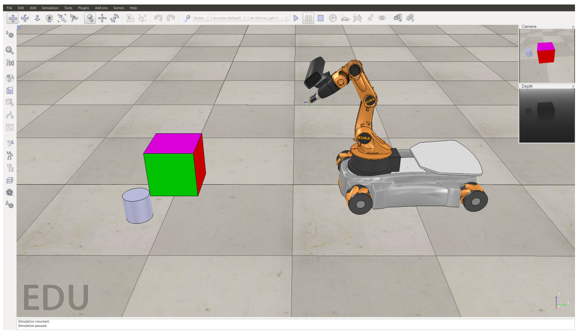


Fig. 4. A scene from V-REP, with the youBot executing the algorithm.

Running the algorithm, we get the results shown in Figures 5-18.

As illustrations, two different cases of movements will be presented. One is orbiting by moving sideways along the y-axis, while changing the desired object distance r_d (case 1), and the other is orbiting by moving upwards along the z-axis (case 2). The user inputs are given functions of time, as shown in Figures 5 and 6.

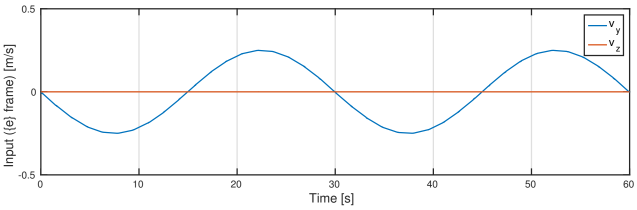


Fig. 5. Case 1: Input during orbit movement, moving in the y-direction of the camera-frame.

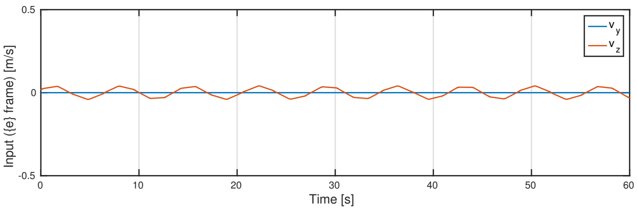


Fig. 6. Case 2: Input during orbit movement, moving in the z-direction of the camera-frame.

We will now see how well the different constraints were satisfied. The first constraint is to keep the required distance to the object, formalized in Equation (7). The corresponding results can be found in in Figures 7 and 8. Given that the approach is reactive, based on the desired user input, we cannot expect that the errors converge to zero, instead, a small remaining lag can be seen in Figure 7.

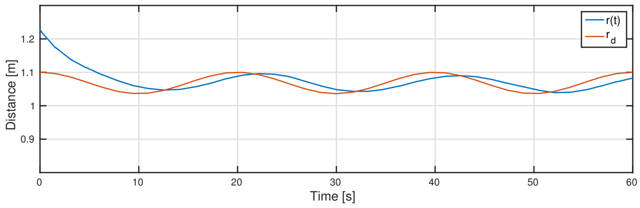


Fig. 7. Case 1: Distance between end-effector and center of object. r_d is the desired distance, and $r(t)$ is the actual distance.

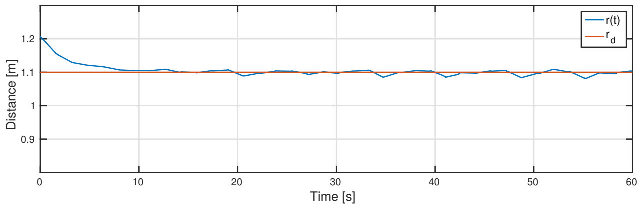


Fig. 8. Case 2: Distance between end-effector and center of object. r_d is the desired distance, and $r(t)$ is the actual distance.

The second constraint is found in Equation (8) and makes sure the object of interest is kept in the center of view. We here present the error as absolute value of an angle. At the start of the simulation, there is a significant error in end effector orientation, as can be seen in Figure 9 and 10. This is significantly reduced, but does not converge to zero. The reason is that this constraint requires motion of both base and the 5 DoF arm and the heavy base is much less precise in its motions, and there is also a modeling error in the simulator model.

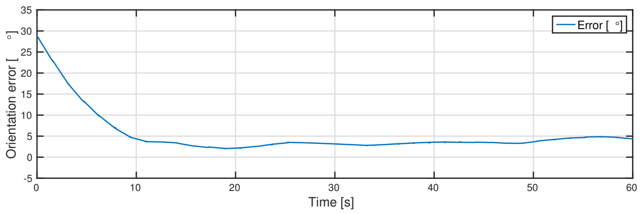


Fig. 9. Case 1: End effector orientation.

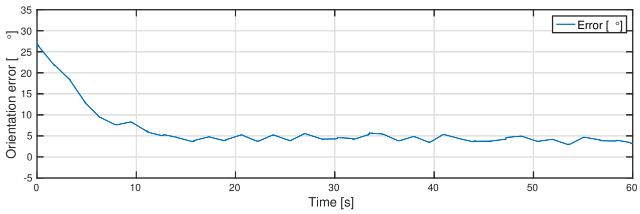


Fig. 10. Case 2: End effector orientation.

The third constraint is found in Equation (9), and makes sure the sensor is not rotating around the line of sight to the object. As can be seen in Figures 11 and 12, this constraint is kept at zero.

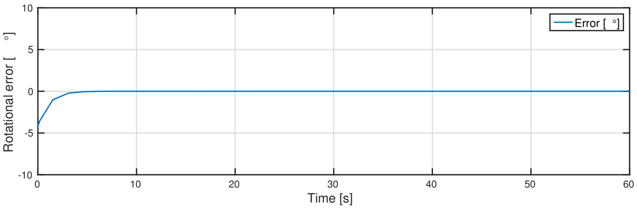


Fig. 11. Case 1: Avoiding rotation around line of sight.

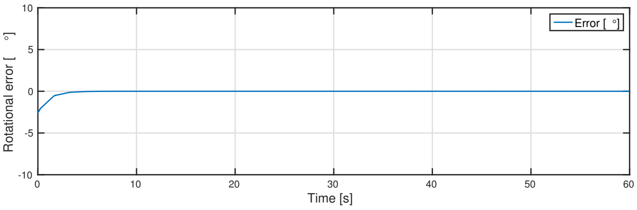


Fig. 12. Case 2: Avoiding rotation around line of sight.

The fourth constraint, found in Equation (10), makes sure that the end effector does not collide with the floor. As can be seen in Figures 13 and 14, this inequality is kept with a margin.

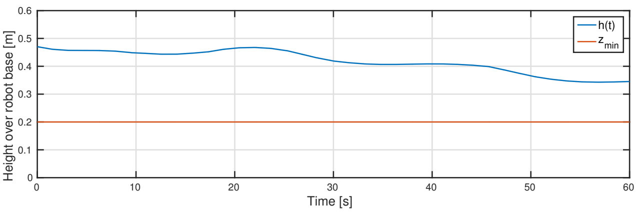


Fig. 13. Case 1: End-effector altitude over base. z_{min} is the minimum allowed height, while $h(t)$ is the actual height.

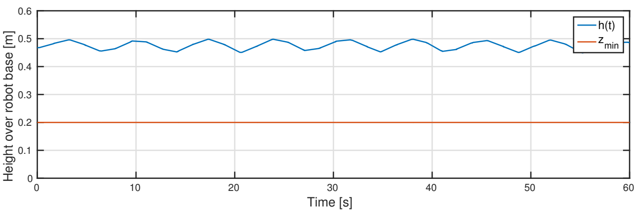


Fig. 14. Case 2: End-effector altitude over base. z_{min} is the minimum allowed height, while $h(t)$ is the actual height.

The fifth constraint, found in Equation (11), makes sure that there are no obstacle collisions. As can be seen in Figures 15 and 16, this inequality is kept with a considerable margin.

Finally, the resulting joint velocities can be found in Figures 17 and 18. As can be seen, they behave reasonably. For case 1, the sine-wave shape is clearly visible.

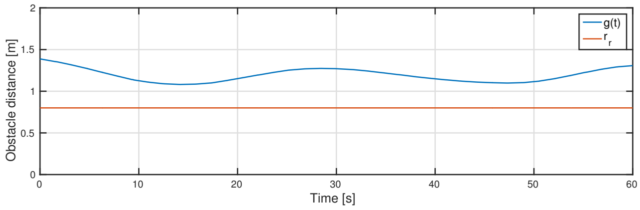


Fig. 15. Case 1: Distance from obstacle. r_r is the desired minimum distance, $g(t)$ is the distance to the obstacle.

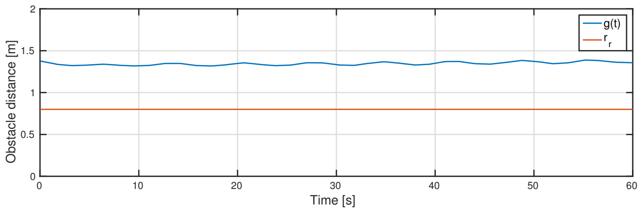


Fig. 16. Case 2: Distance from obstacle. r_r is the desired minimum distance, $g(t)$ is the distance to the obstacle.

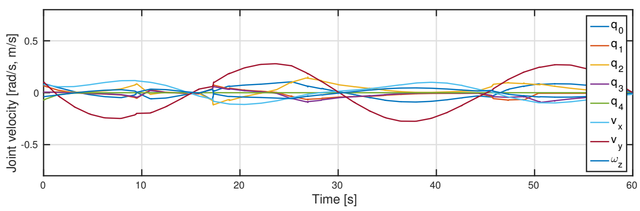


Fig. 17. Case 1: Joint velocities during orbit movement. q_0-q_4 are arm joints, v_x,v_y are translation and v_z rotation of base.

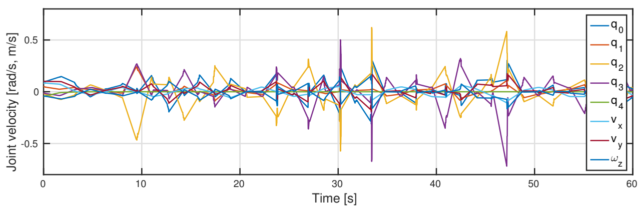


Fig. 18. Case 2: Joint velocities during height movement. q_0-q_4 are arm joints, v_x,v_y are translation and v_z rotation of base.

VI. Conclusions

In this paper we presented an approach for realizing the *orbit object* control mode on a teleoperated mobile manipulator. *Orbit object* is a key function in most 3D design softwares, enabling architects and designers to efficiently manipulate and explore virtual objects.

We believe that this *object centric* function would also provide a strong complement to the *robot centric* and *world centric* control modes described in the robot teleoperation literature.

Using a constraint based framework, we show how to implement *orbit object* on a mobile manipulator, and use V-REP simulations to illustrate the approach.

In the simulation environment the holonomic properties of the youBot has been used, specified through the Jacobian. Though not presented here, it is expected that modifying the Jacobian is sufficient to apply the algorithm to a non-holonomic platform.

APPENDIX: DERIVATIONS OF CONSTRAINTS

In this section we describe detailed derivations of the constraints.

A. Derivative of constraint f_1

Rewriting the constraint into the square root of a scalar product, the constraint follows from the product rule. We rewrite $\mathbf{p}_o^w - \mathbf{p}_e^w$ as $\mathbf{p}_o^e$, for brevity.

$$f_1 := ||\mathbf{p}_o^e||_2 = \sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e},$$

$$\frac{\partial}{\partial q}\sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e} = \frac{1}{2}(\mathbf{p}_o^{e\top}\mathbf{p}_o^e)^{-1/2}(\frac{\partial \mathbf{p}_o^{e\top}}{\partial q}\mathbf{p}_o^e + \mathbf{p}_o^{e\top}\frac{\partial \mathbf{p}_o^e}{\partial q}).$$

Rewriting the sum by taking the transpose of the first term, we have that

$$\frac{\partial}{\partial q}\sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e} = \frac{\mathbf{p}_o^{e\top}\frac{\partial \mathbf{p}_o^e}{\partial q}}{\sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e}}.$$

We now use the fact that we are interested in the translational part, and also that the object is stationary. Thus the derivative of $\mathbf{p}_o^e$ is the Jacobi matrix of the robot base and end-effector, so we arrive at

$$\frac{\partial f_1}{\partial q} = -\frac{\mathbf{p}_o^{e\top}J_t}{\sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e}}.$$

B. Derivative of constraint f_2

By similar calculations as above, we arrive at

$$\frac{\partial f_2}{\partial q} = \frac{\partial \mathbf{p}_o^e}{\partial q}^\top \mathbf{e}_x^e + \mathbf{p}_o^{e\top}\frac{\partial \mathbf{e}_x^e}{\partial q} - \frac{\mathbf{p}_o^{e\top}\frac{\partial \mathbf{p}_o^e}{\partial q}}{\sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e}},$$

where $||\mathbf{e}_z^e||$ and it's derivative is one which simplified the last part of the expression. The final piece needed for this derivative is an expression for $\frac{\partial \mathbf{e}_x^e}{\partial q}$, which can be obtained using the skew-symmetric matrix [20][Ch. 4], to arrive at

$$\frac{\partial e_x^e}{\partial q} = -S(R_e^w e_x^e)J_\omega^b.$$

This leads to the derivative

$$\frac{\partial f_2}{\partial q} = -(R_e^w \mathbf{e}_x^e)^\top J_t - \mathbf{p}_o^{e\top}S(R_e^w \mathbf{e}_x^e)J_\omega + \frac{\mathbf{p}_o^{e\top}J_t}{\sqrt{\mathbf{p}_o^{e\top}\mathbf{p}_o^e}}.$$

C. Derivative of constraint f_3-f_6

Constraints f_3-f_6 are derived similar to f_1-f_2, and thus we omit the details.

ACKNOWLEDGMENT

The authors gratefully acknowledge funding under the European Union's seventh framework program, under grant agreements FP7-ICT-609763 TRADR.

REFERENCES

[1] R. R. Murphy, "Human-robot interaction in rescue robotics," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 2, pp. 138–153, 2004.

[2] J. Richer and J. L. Drury, "A video game-based framework for analyzing human-robot interaction: characterizing interface design in real-time interactive multimedia applications," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on human-robot interaction*. ACM, 2006, pp. 266–273.

[3] J. A. Atherton and M. A. Goodrich, "Supporting Remote Manipulation with an Ecological Augmented Virtuality Interface," *(unknown)*, 2010.

[4] J. Y. C. Chen, E. C. Haas, M. J. S. M. Barnes, C. P. C. Applications, and R. I. T. on, "Human Performance Issues and User Interface Design for Teleoperated Robots," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, Jan. 2007.

[5] H. A. Yanco and J. L. Drury, "Rescuing interfaces: A multi-year study of human-robot interaction at the aaai robot rescue competition," *Autonomous Robots*, vol. 22, no. 4, pp. 333–352, 2007.

[6] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 927–941, 2007.

[7] A. Kelly, N. Chan, H. Herman, D. Huber, R. Meyers, P. Rander, R. Warner, J. Ziglar, and E. Capstick, "Real-time photorealistic virtualized reality interface for remote mobile robot control," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 384–404, 2011.

[8] M. Micire, J. L. Drury, B. Keyes, and H. A. Yanco, "Multi-touch interaction for robot control," in *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 2009, pp. 425–428.

[9] B. Larochelle and G. Kruijff, "Multi-view operator control unit to improve situation awareness in usar missions," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 1103–1108.

[10] P. Ögren, P. Svenmarck, P. Lif, M. Norberg, and N. E. Söderbäck, "Design and implementation of a new teleoperation control mode for differential drive ugvs," *Autonomous Robots*, vol. 37, no. 1, pp. 71–79, 2014.

[11] S. Muszynski, J. Stuckler, and S. Behnke, "Adjustable autonomy for mobile teleoperation of personal service robots," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 933–940.

[12] M. Mast, Z. Materna, M. Španěl, F. Weisshardt, G. Arbeiter, M. Burmester, P. Smrž, and B. Graf, "Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 183–202, 2015.

[13] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 1–8.

[14] E. You and K. Hauser, "Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input," in *Robotics: science and systems*, vol. 7, 2012, p. 354.

[15] M. Frejek and S. B. Nokleby, "A methodology for tele-operating mobile manipulators with an emphasis on operator ease of use," *Robotica*, vol. 31, no. 03, pp. 331–344, 2013.

[16] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 1993, pp. 76–82.

[17] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 953–966, 2004.

[18] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 2939–2944.

[19] Y. Wang, F. Vina, Y. Karayiannidis, C. Smith, and P. Ogren, "Dual arm manipulation using constraint based programming," in *IFAC World Congress, Cape Town, South Africa*, 2014.

[20] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

2016 IEEE International Symposium on
Safety, Security, and Rescue Robotics (SSRR)
EPFL, Lausanne, Switzerland, October 23-27, 2016

3D Localization, Mapping and Path Planning for Search and Rescue Operations

R. Dubé*, A. Gawel*, C. Cadena, R. Siegwart
Autonomous Systems Lab - ETH Zurich, Switzerland

L. Freda*, M. Gianni
ALCOR Lab, DIAG - Sapienza University of Rome, Italy

***Abstract*—This work presents our results on 3D robot localization, mapping and path planning for the latest joint exercise of the European project "Long-Term Human-Robot Teaming for Robots Assisted Disaster Response" (TRADR)[1] . The full system is operated and evaluated by firemen end-users in real-world search and rescue experiments. We demonstrate that the system is able to plan a path to a goal position desired by the fireman operator in the TRADR Operational Control Unit (OCU), using a persistent 3D map created by the robot during previous sorties.**

I. INTRODUCTION

Teams of autonomous mobile robots have the potential to reduce human risks during disaster response as well as the associated costs [1]. Different levels of robot autonomy are required in order to effectively support a rescue squad performing high-level tasks such as exploring the disaster area, detecting victims and taking chemical samples. Moreover, long-term operation of robotic platforms is desired for humans and robots to collaborate over several days of disaster intervention. To this end, building and maintaining a persistent representation of the environment, accurate localization, and efficient path planning are fundamental prerequisites.

Prior to this work, a SLAM strategy based on Iterative Closest Point (ICP) for the robotic platform considered in this work was proposed in [2]. While providing precise local reconstruction of an environment, this technique can not improve the map in the event of place recognition. The localization algorithm presented in this paper is therefore based on the pose-graph SLAM strategy as described in [3].

The 3D path planning and navigation methods presented in this paper are based on the works [4, 5, 6]. The underlying modules provide functionalities such as real-time point cloud segmentation and traversability analysis. A randomized A* approach is applied on the current terrain structure interpretation.

In the remainder of this report, we concisely describe the localization, mapping and path planning systems and present the results of experiments with firemen end-users, at the latest TRADR Joint Exercise (TJEx).

II. SYSTEM DESCRIPTION

While the TRADR system comprises an integrated framework spanning from low-level perception functionalities to high-level reasoning, in this work we focus on presenting the latest advances in the integrated SLAM and path planning.

*R. Dubé, A. Gawel and L. Freda contributed equally to this work.

[1]This research is supported by EU-FP7-ICT-Project TRADR 609763. http://www.tradr-project.eu/

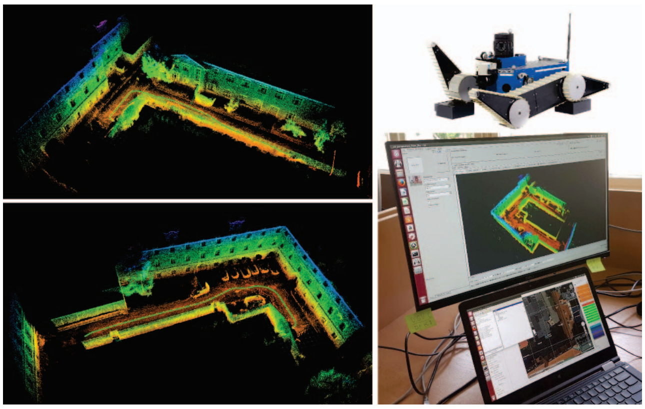


Figure 1: *Top-left*: 3D laser map generated in a first sortie on mission day 1. *Bottom-left*: Map generated during a second sortie on mission day 2. *Top-right*: TRADR UGV equipped with multiple encoders, an IMU and a rotating laser-scanner. *Bottom-right*: TRADR OCU displaying the merged map.

A. Pose-graph based localization, mapping and map merging

The proposed SLAM system relies on a pose-graph optimization approach [3]. This framework computes a Maximum A Posteriori (MAP) estimate of a trajectory of robot poses $\boldsymbol{c}(t_i) \in SE(3)$ collected at times $\{t_i\}_{i=0}^N$ by optimizing a negative log-posterior E (aka error function) which sums over constraints $\boldsymbol{\Theta}(\boldsymbol{c}_{i,j}) = \boldsymbol{e}_{i,j}^T \boldsymbol{\Omega}_{i,j} \boldsymbol{e}_{i,j}$, with $\boldsymbol{e}_{i,j} = \boldsymbol{z}_{i,j} - \tilde{\boldsymbol{z}}_{i,j}(\boldsymbol{c}_{i,j})$, and $\boldsymbol{z}_{i,j}$ the observation, $\tilde{\boldsymbol{z}}_{i,j}(\boldsymbol{c}_{i,j})$ the prediction and $\boldsymbol{\Omega}_{i,j}$ the information matrix. In the present system the function E is obtained by combining together *(i)* odometry constraints $\boldsymbol{\Theta}_O(\boldsymbol{c}_{i,j})$, fusing wheel encoders and IMU data along the lines of [7], and *(ii)* laser scan-matching constraints $\boldsymbol{\Theta}_S(\boldsymbol{c}_{i,j})$, from ICP matching the current scan against all previous scans within a sliding time window $[t-w, t] \subset \mathbb{R}$ where t is the current time and w is the chosen fixed time width. Let $\boldsymbol{c}(t_1{:}t_2)$ denote the sequence of robot poses acquired in the time interval $[t_1, t_2] \subset \mathbb{R}$ and $\mathcal{C}_O$ and $\mathcal{C}_S$ respectively the set of pairs of timestamps for which odometry and scan-matching constraints exist over that time interval. The robot trajectory $\boldsymbol{c}(t)$ is finally estimated by incrementally minimizing the error

$$E(\boldsymbol{c}\,(t-w{:}t)) = \sum_{<t_i,t_j>\in\mathcal{C}_O} \boldsymbol{\Theta}_O(\boldsymbol{c}_{i,j}) + \sum_{<t_i,t_j>\in\mathcal{C}_S} \boldsymbol{\Theta}_S(\boldsymbol{c}_{i,j}) \tag{1}$$

on the sliding window. This approach is flexible and allows registration of further error terms, e.g., loop closure constraints.

The global 3D map is the result of (i) projecting individual laser scans from their respective recording locations $\boldsymbol{c}(t_i)$ into a world reference frame and (ii) optionally applying point cloud post-processing filters, e.g., downsampling. For removing dynamic objects in the map, the system offers a probabilistic filtering based on octomaps.

The SLAM approach furthermore allows the reuse of previously recorded maps in subsequent sorties. The current approach for map merging aligns the robot's previous world reference frame with the current one at starting time and loads the previous map. The robot then continues mapping by constructing a new pose graph from its starting location. Both the loaded map and the updates are accessible to other robot modules, e.g., the path planner, giving rise to persistent use of multi-sortie information.

B. *Path planning on built maps*

The navigation module accepts as input both online registered point clouds and maps built in past sorties. When new sensory data is available and if substantial changes occurred in the map, the structure interpretation of the point cloud is updated. As a first step, the point cloud is filtered and *geometric features* such as normals and surface curvatures are computed. Then, segmentation is performed and clusters are labeled according to geometrical constraints applied to surface normal directions, mean curvatures and 3D-coordinates of points. This results in a classification of the environment in regions such as *walls*, *terrain*, *surmountable obstacles* and *stairs/ramps* [4, 5]. Traversability is then computed as a *cost function* taking into account the point cloud classification and the local geometric features [4, 5] (such as obstacle clearance, terrain roughness and point cloud density).

Path planning is performed both on global and local scales. Given a set of waypoints as input, the *global* path planner checks the existence of a traversable path joining them. Once a solution is found, a *local* path planner drives the robot towards the closest waypoint by continuously replanning a feasible path in a local neighbourhood in order to take into account possible dynamic changes in the environment. On both global and local scale, the connectivity of the traversable terrain is captured by using a sampling-based approach. In particular, a tree is expanded in the configuration space by using a randomized A* approach [4, 5].

III. INTEGRATED SCENARIO EXPERIMENTS

The full system was evaluated at the latest TJEx experiments where firemen end-users performed a search and rescue mission by teleoperating two TRADR UGVs. Amongst other sensors, these skid-steered vehicles are equipped with a 360° spherical camera and a rotating laser scanner as shown in Fig. 1, top-right.

An initial sortie was executed during the first mission day resulting in the map depicted in Fig. 1. On the second mission day, a second sortie was performed with a different robot, extending the map generated during the first sortie. The resulting merged map was displayed to the end-users in the command post through the TRADR OCU.

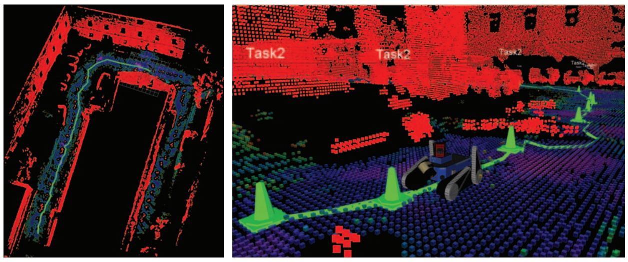


Figure 2: *Left*: Segmentation of the merged map into obstacles (red) and traversable regions (blue); a globally planned path (green line) is shown on the traversable region. *Right*: An example of planned path (green line) joining a set of waypoints (green traffic cones) selected by the end-user, directly on the traversability map.

During each mission, the end-users were able to identify points of interest and mark them as navigation waypoints on the traversability map (Fig. 2, left). Each set of selected waypoints was fed into a *task queue* managed by the global path-planner, which was always able to successfully compute a traversable path (Fig. 2, left). At execution time, the local path-planner and the trajectory control safely drove the vehicle along the planned paths (Fig. 2, right) by performing a continuous replanning in order to manage possible low-dynamic changes occurring in the environment. Autonomous waypoint inspection was successfully performed allowing end-users to detect victims and possible gas leaks.

IV. CONCLUSION

In this report we have shown the results of the mapping and path planning systems on the latest TJEx. A full map was obtained from different sorties by tele-operating the robots for exploration of the disaster area. This allowed to reduce the level of tele-operation in posterior sorties by automatically planning suitable paths on the built map for further inspection of interest points. Remarkably, the firemen were able to fully operate the robots by using the TRADR OCU which demonstrated the robustness and reliability of our 3D localization, mapping, and path planning systems. These results bring the goal of effective human-robot teaming closer to reality.

REFERENCES

[1] D. Tardioli, *et al.*, "Robot teams for intervention in confined and structured environments," *Journal of Field Robotics*, 2015.

[2] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[3] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.

[4] G. Kruijff, , *et al.*, "Designing, developing, and deploying systems to support human–robot teams in disaster response," *Advanced Robotics*, vol. 28, no. 23, pp. 1547–1570, 2014.

[5] M. Menna, M. Gianni, F. Ferri, and F. Pirri, "Real-time autonomous 3d navigation for tracked vehicles in rescue environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.

[6] M. Gianni, F. Ferri, M. Menna, and F. Pirri, "Adaptive robust three-dimensional trajectory tracking for actively articulated tracked vehicles," *Journal of Field Robotics*, pp. n/a–n/a, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21584

[7] V. Kubelka, *et al.*, "Robust data fusion of multimodal sensory information for mobile robots," *Journal of Field Robotics*, vol. 32, no. 4, pp. 447–473, 2015.