Towards a Semi-Autonomous Robot Platform for the Characterisation of Radiological Environments

DOI: 10.1109/SSRR56537.2022.10018668
Handle: http://hdl.handle.net/1942/39922

Towards a Semi-Autonomous Robot Platform for the Characterisation of Radiological Environments

David De Schepper[1], Ivo Dekker[1,3], Mattias Simons[2], Lowie Brabants[2], Wouter Schroeyers[2], and Eric Demeester[1]

***Abstract*—During the last decades, the (partial) automation of tasks during the dismantling and decommissioning of potentially nuclear contaminated environments has become of emerging interest for e.g. homeland security, disaster response, continuous maintenance, and dismantling and decomissioning activities. Nowadays, the nuclear scene is mostly dominated by manual labour. Radiation protection officers have the task of characterising an environment, which is often unknown a priori, before any dismantling and decomissioning activity can take place. Besides the potential involved health risks, going from radiation disease to an increase in the risk of cancer, this important preliminary task is very time-consuming and prone to errors concerning the taken measurements, storage and post-processing of the recorded data. To minimise the disadvantages, this paper presents the design and development of a proof-of-concept semi-autonomous, ground-based mobile manipulator robot ARCHER (Autonomous Robot platform for CHaractERization) suited for radiological monitoring purposes. Besides the mechanical and electrical overview of the design of the mobile manipulator, this paper describes the software tools used to build and deploy the robot. In addition, this paper describes the results of several in-situ laboratory experiments where the mobile manipulator platform is asked to perform a radiological scanning task on a wall.**

***Index Terms*—Robotics for nuclear power plants; Mobile manipulator; Radiological mapping**

I. Introduction

Operational nuclear power reactors are progressively reaching the end of their operating life nowadays. As a result of this, prospects are high that the amount of activities for the purpose of nuclear dismantling and decommissioning will increase in the near future [1], [2].

[1] KU Leuven, Faculty of Engineering Technology, Department of Mechanical Engineering, Campus Diepenbeek, ACRO Research Group, Wetenschapspark 27, 3590 Diepenbeek, Belgium.

[2] Hasselt University, Faculty of Engineering Technology, CMK, Nuclear Technology Centre, Agoralaan, gebouw H, 3590 Diepenbeek, Belgium.

[3] Core Lab ROB, Flanders Make @ KU Leuven, Belgium

Corresponding author: david.deschepper@kuleuven.be

The authors would like to acknowledge the financial support of the Belgian Federal Public Service (FOD) Economy, Energy transition fund towards the ARCHER project. The authors would also like to thank Vlaams Agentschap Innoveren & Ondernemen (VLAIO) for granting Ivo Dekker's Baekeland mandate HBC.2020.2884, also facilitating this research.

The first step towards defining an effective dismantling and decommissioning strategy, is to possess a priori knowledge of the potential radioactive contamination in the environment, and, more importantly, the radioactive of potential sources of high radiological contaminations, often mentioned in literature as nuclear hot spots. This a priori knowledge is of high importance during decontamination operations, aimed at minimising the amount of added radioactive waste, an effective strategy in keeping decontamination costs as low as possible [3]. Furthermore, extensive characterisation of the environment that will be dismantled is a crucial step in keeping operator dose exposure as low as possible during the entire decontamination process. This early detection of radiological sources, and subsequent early removal of these high dose rate locations, will ensure strict compliance with the good practice of keeping radiological dose exposure as low as reasonably achievable, according to the ALARA standard [4], [5].

Nowadays, the nuclear scene is mainly characterised by manual labour. To perform in-the-field measurements (exposure and contamination measurements) for instance, radiation protection officers (RPOs) are currently employed. This comes with some severe drawbacks:

1) the manual process of acquiring measurement data is very time-consuming. In practice, the duration of this process can last ten minutes or a few hours, depending on the size and type of the environment;
2) a priori screening of environments is subject to a variety of uncertainties. Examples of these types of uncertainties can be found in the type of sensor used to obtain the observation data, the angle between the measurement device and the environment (e.g. an ideal measurement is perpendicular to the object to be measured), or the storage of the obtained data used for post processing of the measurements (e.g. radiological heat map creation);
3) due to the potential high dose rates in the environment to be monitored, the physical and mental state of the RPO personnel needs to be monitored at a frequent rate to prevent potential, future health

damage.

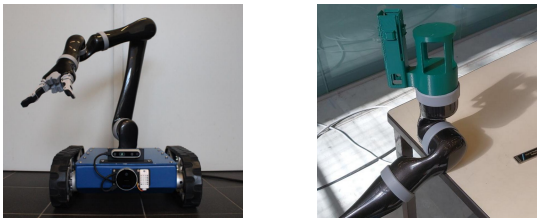
Robotic or semi-autonomous assistance in terms of a semi-autonomous robotic platform during these processes can lead to minimising the operator risks and keeping the operator costs as low as possible. Over the past decades, research has been performed and research projects have been carried out towards the use of robotic systems in the nuclear scene and the construction and development of these systems. The first generation of robotic systems employed in the nuclear scenes were deployed in the 1980s. Most of these robot platforms were teleoperatedly driven [6]. Nowadays, new research has been focusing on fully autonomous robots, capable of characterising an environment in a (semi-)autonomous manner [7]–[10]. Bird et al. [10] described the design and development of a mobile, autonomous, radiological-monitoring robot called CARMA. It was the first proof-of-concept robot that was designed to be deployed into an active nuclear site in Europe. In the proof-of-concept experiments, the robot showed it was capable of identifying and localising a single source of α radiation at the Sellafield nuclear site. Ducros et al. [8] presented the implementation of a track-driven mobile robot RICA developed to sample and characterise radiological information (γ radiation) in the nuclear field. Besides the standalone usage of the mobile robot, a robotic arm can be fitted on top of the mobile platform, exploring the benefits of the mobile platform and the robotic arm (manoeuvrability and reachability). Besides the development of robots suited for maintenance and inspection of nuclear sites, research towards adopting robots in the nuclear scene with cognitive skills is booming [11]–[14]. Deterministic methods to localise radiation source(s) in a(n) (un)known environment(s), e.g. through the use of regression techniques and heat map generation based on a collection of measurements or a data set, have been introduced, as well as Bayesian techniques (as known within the field of probabilistic robotics) are being used to localise radiation source(s) since a couple of years [15]. This resulted for instance in various (semi-)autonomous approaches for dismantling purposes at Fukushima [7], [16]–[18].

This paper describes the design and development of the proof-of-concept ARCHER robot platform, an abbreviation for Autonomous Robot platform for CHaractERization, suited for exploration, inspection, and characterisation of potentially nuclear contaminated environments. The robot is designed and developed by a Company-University consortium consisting of Tecnubel[1], a Belgian-German company with main activities in the decommissioning and logistics of nuclear waste; MAGICS®[2], a technology company which provides electronic solutions for intelligent sensing and smart robotics for extreme environments; KU Leuven's research group ACRO, part of KU Leuven's robotics group, and Hasselt University's Nuclear Technology Centre (NuTeC research group). The remainder of this paper is organised as follows. Section II initiates with the design and development specifications of the ARCHER platform. Besides the mechanical and electrical design, the communication and software implementation is discussed. Section III discusses the performed proof-of-concept experiments conducted with the developed platform. Finally, section IV concludes with a broad summary of this work and deliberates tracks for future work.

II. Hardware Design and Software Implementation

A. Mechanical Design

Mechanically, the ARCHER robot is comprised of a drivable navigation unit on tracks in combination with a six degrees of freedom Kinova Jaco 2 robotic arm, namely a `j2s6s200` model. These robot arms come with a two or three fingered gripper used for manipulation. However, this tool will not be used further in this application and will therefore be replaced with a in-house developed end effector displayed in figure 1. The main hardware components are listed up in table I.



(a) (b)

Fig. 1: The developed ARCHER robot (a) and the end effector holder (b), suited to be mounted as the last link of the kinematic chain of the Kinova Jaco 2 robotic arm. The newly designed tool can be used to mount the Kromek GR1 CZT sensor and the RealSense L515 camera on the end effector of the robotic arm.

TABLE I: Specifications of the ARCHER robot platform

Parameter	Type
Robot platform dimensions (mobile base)	450mm x 400mm x 170mm
Robot arm reach	980mm
Robot platform weight	25kg
Robotic arm	Kinova Jaco 2 6DOF spherical
γ sensor	Kromek GR1 CZT
Navigational sensors	RealSense L515 LiDAR camera Rotary wheel encoders
Onboard processing	Intel UP Squared

[1]http://www.tecnubel.be/

[2]https://www.magics.tech/

B. Electrical Design

The ARCHER platform is being powered by a wired connection to ensure a reliable supply of power. A wired power supply has been chosen over a battery pack to

1) save extra, unneeded weight: By not using a battery pack on or inside the mobile platform, its weight is reduced so that a single operator is able to lift the entire unit in case of any emergency.
2) have a longer cycle time: By limiting the use of a battery pack, the ARCHER platform is able to work an entire work day, i.e. eight hours, without the need of a recharge.

Figure 2 shows an overview of the hardware design of the ARCHER platform together with the data and communication flow between the different entities.

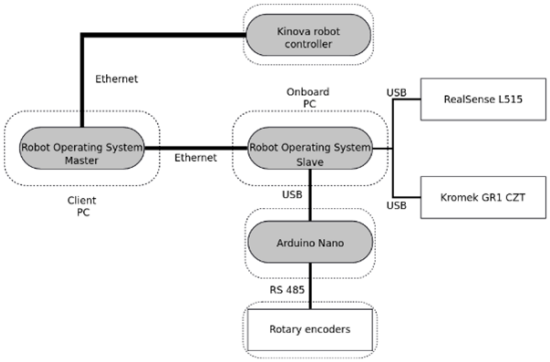


Fig. 2: Schematic overview of the hardware (mechanical, electrical, and communication) architecture of the ARCHER mobile manipulator platform. The architecture consists of four major components (displayed in gray): (i) the client PC where the Robot Operating System (ROS) is enabled as a master, and is used to enable the main software development; (ii) the onboard host PC (Intel UP Squared) where ROS is enabled as a slave. The onboard PC is connected with the client PC using a wired Ethernet connection; (iii) a microcontroller (Arduino Uno) which enables the low-level communication with the rotary encoders, and (iv) the Kinova robot controller which is connected with the client PC using a seperate wired Ethernet connection.

C. Data and communication

Data transfer and communication with the host computer is established via an Ethernet connection. Two seperate Ethernet cables are utilised, where one serves the purpose of reading the Intel Realsense L515 LiDAR camera, the Kromek CZT GR1 sensor and communication with the Intel UP Squared on-board computing unit. The other is solely used for controlling the Kinova arm. The reason for this two-cable setup is due to an issue which occurs when using the manipulator in combination with the sensor and the on-board computer via a singular cable. A gigabit Ethernet switch is used to facilitate all devices using the same Ethernet connection. However, adding the robot to this setup has shown to be non-functional in this setup at this point in time. If the robot is commanded over an Ethernet switch, the ROS driver used will inevitably fail and break connection.

Measurement data received from the Kromek sensing probe is processed on the Intel UP Squared. These spectrum measurements are later combined with the 6-DOF robot pose at the point of measurement for further analysis. RGB image and depth data received from the RealSense L515 LiDAR camera are processed by the Intel UP Squared on-board computing unit, which then transfers it to the host PC. If bandwidth limitations are concerned, it is possible to utilise compressed data from the camera to reduce the load on the Ethernet connection.

D. Software Design

The software running the ARCHER platform has been designed using the open-source Robot Operating System (ROS) as a software framework [19]. The overall software architecture of the ARCHER platform can be separated into three major high-level categories, namely the parts controlling:

1) Mobile navigation
2) Robot arm control and collision-free trajectory generation
3) Kromek spectrum measurements

1) Mobile navigation: One of the tasks that the ARCHER platform should be capable of is to build a geometric model of the environment. This is the process of simultaneous localisation and mapping (SLAM), where a mobile robot uses proprioceptive sensors in combination with exteroceptive sensors to build a map of its environment while still keeping track of its location in an iterative, probabilistic fashion [20]. In order to perform SLAM on the mobile robot, odometry data together with a kinematic model is used to give a better estimation of the position of the robot in the environment. The ARCHER platform consists of two DC powered motors connected, via shafts and a transmission, to its outer tracks, which results into a differentially driven, track-based mobile robot. The control of such a mobile robot is composed of sending a twist command $\boldsymbol{t} = (v, \omega)$, a vector comprising the linear and angular velocity respectively, to the motors. The kinematic profile has been experimentally measured using the encoder output on each powered motor, resulting in the graphical representation of the differential velocity profile of figure 3.

In addition, to ensure that the robot keeps track of its location robustly, the processing of its encoder signals have been tweaked and corrected using the approach of [21], an iterative process in which the odometry errors

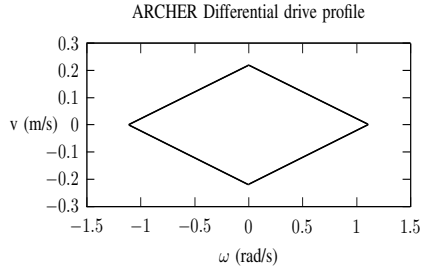


Fig. 3: Differential drive velocity profile for the ARCHER robot platform. The kinematic profile has been experimentally validated using the output of the platform's rotary encoders. The maximum linear velocity of the ARCHER platform amounts $\pm$ 0.22 m/s; the maximum rotational velocity is measured to be $\pm$ 1.11 rad/s.

are calculated and taken into account in defining the kinematic model. Using the rotary encoders and the RGB and depth data of the RealSense L515 camera, the Real-Time Appearance-Based Mapping (RTAB-map) algorithm of [22] has been chosen as a 3D SLAM algorithm for the ARCHER platform. The quality of the generated maps could be further evaluated using the methods proposed in [23]. Together with the ability of building a map of the environment, the ARCHER platform is able to perform autonomous mobile navigation. For this, the `ROS navigation stack`[3] is used together with the `move_base`[4] functionalities for global and local path planning within the Robot Operating System framework.

2) Robot arm control and collision-free trajectory generation: Once the robot has finished its mobile navigation sequence, it becomes stationary and activates its manipulator. The manipulator is then controlled using MoveIt!, a motion planning framework integrated within ROS [24]. First, there is the task of moving from the home position of the robot to the start of the desired scanning sequence, and back to its home pose once the scan is completed, in a collision-free manner. To ensure the system knows its surrounding area to be able to generate collision-free trajectories, a scanning motion is first executed which will make the arm and the L515 camera rotate around the Z-axis of the robot base at varying heights. The depth data acquired by the L515 camera will then be used to generate an Octomap collision representation of the surroundings of the robot. Octomap generates a voxel-based grid around the received depth points which will be treated as collision objects in the simulated robot environment [25]. MoveIt! will take these voxels into consideration during trajectory generation. The movement to the start of a scan sequence and back home afterwards is planned using a Rapidly exploring Random Trees (RRT) planning algorithm. As explained in [26], the RRT planner works by "growing" two random trees, where one is rooted in the starting state and the other in the desired end state. This results in two separate trees of collision-free states. The planner then proceeds to look for states which are common to both trees, which it will combine to form the final trajectory. RRT is chosen as it is a single-query planner which explores the configuration space for each individual planning request and thus having the benefit that the planner will find trajectories (given one exists) in changing environments, which is mandatory for this application.

Second, the robot has to perform the motion of the desired scanning sequence. Scanning sequences for flat surfaces are generated automatically by plane segmentation on the found point cloud data received from the L515 sensor. Using the `Multiplane_SAC_Segmentation` algorithm offered by [27], [28], multiple different planes can be extracted from the point cloud data. These planes are surrounded by a polygon. Using an `Interactive_Polygon_Marker` also supplied by the package made by [27], the different polygons of the found planes can be selected by the user, which will publish the information of the selected polygon on a ROS topic. Next, this polygon's data are transferred to a Python script subdividing the polygon into a grid of points. It does this by fitting the smallest possible rectangular bounding box around the polygon. Next, the code subdivides this rectangle into a set of equidistant points along the directions of two perpendicular sides of the bounding box. Lastly, the code checks which of the generated points fit inside the original polygon and only saves the points which adhere to this constraint. This array of points is then translated into robot poses. By using singular value decomposition (SVD), the position and orientation of each pose is determined, making sure the sensor probe will always be oriented perpendicular to the plane by aligning it with the plane's normal direction. Finally, this list of poses is executed by the robot by following a Cartesian path from point to point. The Cartesian movement is used to force the robot to move in a straight line in between each point. If these motions would be planned using a RRT planner as well, it could occur that the planner would find long, convoluted trajectories due to how the RRT planner operates. The downside of using the Cartesian motion is that the robot will no longer actively avoid collisions if one would occur. Instead, it will simply refuse to execute the motion rather than move into a collision state.

[3]http://wiki.ros.org/navigation

[4]http://wiki.ros.org/move_base?distro=noetic

3) Spectrum measurement: Besides the gathering of geometrical information using rotary wheel encoders mounted on the shaft of the DC motors and RGB and depth information from a LiDAR camera, the ARCHER platform is capable of obtaining radiological information from the environment by using data extracted from the Kromek GR1 CZT sensor[5]. To incorporate the radiological data stream in the ROS software framework, a ROS driver has been developed to communicate with Kromek sensors [29]. The ROS driver consists of two entities, a ROS publisher and a ROS service, fused together in one driver node, as depicted in figure 4. The publisher part of the driver node reads out data from the application layer and publishes an implemented count rate message at a fixed frequency. This frequency can be set by the user through the ROS parameter server. Alongside the publishing function of the driver node, a ROS service is implemented. This service listens to other ROS nodes which try to request data from the service. When a node is requesting data, the driver stops publishing data until the service request is being handled and a response is generated. In the service request, the measuring time to obtain a spectrum is embedded. The response that the ROS service generates is an obtained radiological spectrum, often obtained at a fixed position. When the request is handled, the driver node starts publishing its data again in a continuous manner. In the context of the software framework built around the ARCHER platform, radiological data can be accessed in the form of a count rate, expressed in counts per second, and in the form of a spectrum message, comprised as a 4096-dimensional array, where each entry of the array represents an energy value for a specific radioisotope. The data can be accessed continuously by subscribing to the ROS specific message topic, or using a service request.

The latter one is used in combination with the software for generating collision-free trajectories and for controlling the Kinova robotic arm. A ROS node will communicate with the Kromek ROS driver through the `service request-reply` protocol. Within the `request` message, the specific amount of time to capture spectral data is specified. The `reply` message contains the spectral data. When the Kromek ROS driver responds with the radiological data, two log files are simultaneously created for post processing purposes. In the first log file, the kinematic data is stored containing the pose of the mobile manipulator, comprised as a three dimensional mobile robot pose, and a six dimensional robot arm pose containing the 3D Cartesian position and the 3D orientation comprised in a quaternion notation. In the second saved log file, the radiological data will be stored. At each acquired robot measurement point, the full spectrum is stored together with its accompanied time stamp.

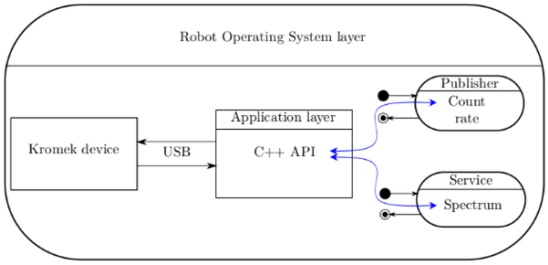


Fig. 4: A graphical overview showing the functionalities of the `kromek-ROS` package. The ROS layer consists of two interfaces (a ROS publisher and a ROS service node) which interacts with the application layer. The application layer consists of the low-level C++ API, where communication with the device over a serial USB connection is established.

III. Experiments

This section gives a brief overview of the conducted proof-of-concept experiments regarding the testing and development of the ARCHER mobile manipulator platform. To get a clear overview, this section has been divided into three subsections regarding the experiments on (i) mobile navigation and mapping (SLAM), (ii) robot arm control and collision avoidance based on selected segmented plane data and (iii) the measuring and data processing of radiological spectrum measurements. Footage of these experiments can be found via the YouTube channel of our research group[6].

A. Mobile navigation

The task of the ARCHER platform is to create a map of an unknown environment, while simultaneously keeping track of its location in that built environment. To test the mobile navigation part in a proof-of-concept experiment, an office environment has been chosen. Figure 5 depicts several key frames from this proof-of-concept experiment. The robot platform begins its SLAM routine at its starting location, depicted in figure 5a. The robot is being teleoperated by an operator using a joystick as a controller. At a second selected key frame, as shown in figure 5b, the robot has mapped half of the entire environment area. The created map shows that the walls are mapped well and are perpendicular. The ARCHER platform uses the compressed data stream of the RealSense L515 LiDAR camera to build a map of the environment. Figure 5c is showing the

[5]The product can be found via following link https://www.kromek.com/product/gr1/.

[6]https://youtu.be/yP0H3m0rrAk

ARCHER platform while traversing an obstacle (a cable tray mounted on the floor). As can be noticed in the figure, map building of the unknown environment is not halted by traversable obstacles. After the SLAM routine, the created map of the environment can be saved, and can be reloaded to perform (semi-)autonomous navigation as discussed in section II-D1.

B. Robot arm control and collision avoidance

To test the collision free trajectory generation, the robot platform is placed in differing environments using boxes as collision objects. These boxes will be randomly oriented around the robot in each environment. An example of a testing environment with its respective Octomap representation can be seen in figure 6. Next, the operator will move the robot arm around in the environment, attempting to bring it into a collision state to verify how well the Octomap collision geometry matches the real-life environment. The voxel size of the Octomap representation should also be chosen carefully. If the resolution of the voxels is too high (small voxels) the computer cannot process the data in time. If, on the other hand, the resolution is too small (large voxels), details of the environment can be lost. After testing, it was found that voxels of size 40 mm at an update rate of 6 Hz yield accurate results while remaining manageable for the computer regarding processing power. To prevent large Octomap sizes, the map data is only stored during the scanning application while the robot remains in its stationary position. Once it moves to a new scanning position, the previous Octomap is forgotten.

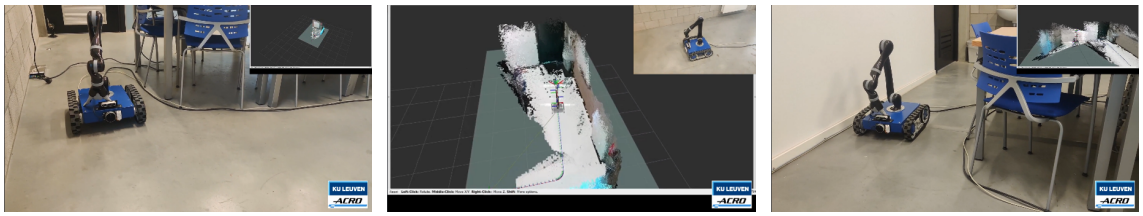
C. Plane segmentation and spectrum measurement

The spectrum scanning motion is tested by placing a detectable surface in the camera frame of the robot. The position and angle of this surface with respect to the camera frame is changed for each attempt. As shown by figure 6c, the operator proceeds to select the detected surface in the RViz visualisation environment to generate a grid of points for scanning. Once the grid is generated, the end effector will move to the first point of the grid in a collision-free manner using a RRT planner cfr. section III-B, where it will call the `ROS-service` responsible for handling the Kromek spectrum measurement. This will cause the end effector to wait in place until the `ROS-service` receives a response, at which it will move to the next point where the process is repeated until the entire grid is covered. Multiple tests show that the robot is capable of performing the scanning motion for surfaces at any angle while keeping the Kromek CZT sensor perpendicular to the surface to be scanned, with the assumption that the surface can be detected and the desired points are within the reaching limits of the robotic arm. As stated earlier, the measured data is stored in two separate log files. These log files can now be used to further analyse the radiological data of the environment. A post processing example of this could be to generate a heat map of the spectrum to map out the hot spots in its measured environment. An example image is shown in figure 7. Although this picture depicts a 2D representation, the same principles could easily be translated to a 3D representation as future work.

IV. Conclusions and future work

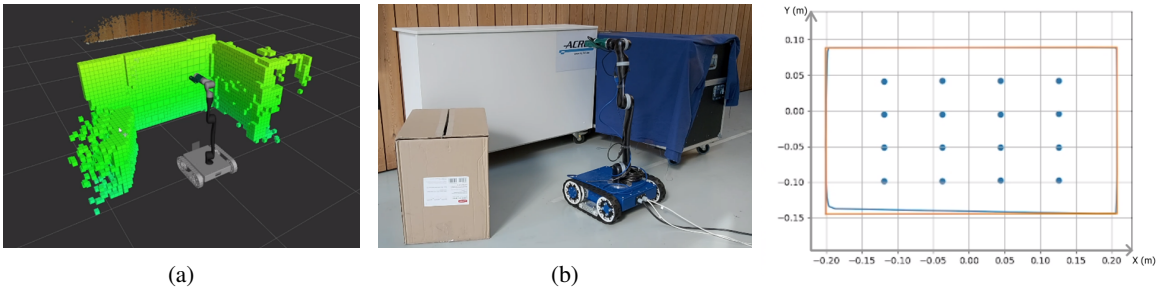
This paper presented the ARCHER robot platform, a (semi)-autonomous, ground-based mobile manipulator robotic system designed and developed towards radiological monitoring purposes in potential nuclear contaminated environments. Due to the various risks associated with the manual process of entering an unknown, potentially contaminated environment, this proof-of-concept robot platform could lead to a minimisation of the risks and costs that come along with the dismantling and decommissioning, and lifelong maintenance of nuclear power sites. Besides the mechanical and electrical specifications, this paper gave an overview of the data and communication flow, as well as an overview of the specific software components and architecture. Furthermore, this paper illustrated a few case study examples what the ARCHER platform is capable of. In addition to building a map in a (semi)-autonomous manner using RTAB-Map's 3D SLAM algorithm, the ARCHER robot is capable of planning and executing collision-free trajectories for the mobile base. On top of this, the ARCHER platform is capable of using the robotic arm's maneuverability in the case of scanning, identifying, and characterising parts of the environment based on radiological data coming from an industrial, radiological sensing device. In this subtask, the robot arm can be adopted to identify and extract planes from the point cloud data. Using the segmentated planes' information, an automatic, collision-free scanning procedure can be called where different sampling points are generated and reached to perform nuclear data extraction. This data will be stored for post-processing purposes where heat maps can be generated by the radiation protection operators.

Tracks for future work concern the upgrading of the segmentation task from planes to cylindrical-shaped objects. Besides walls, cylindrical-shaped objects, e.g. gas cylinders, tanks, and large buckets are the most common objects in nuclear environments. For this type of objects, the segmentation task using the point cloud data will have to be upgraded. In addition, real life experiments are planned where the geometric and radiological mapping approaches will be quantitatively and qualitatively evaluated. Furthermore, the use of finite state machines and behaviour trees for scheduling between different tasks will be explored [30]. Additionally,



(a) (b) (c)

Fig. 5: Selected key frames from executing the experiments involving the building of an environment model using RTAB-Map's 3D SLAM approach. An office room has been chosen as a testing environment for the mobile navigation part. Figure 5a is depicting the robot in its begin position; figure 5b is showing a visualisation of the built map at a specific time stamp. As can be seen, an entire part of the room has been mapped. Figure 5c visualises the robot while the ARCHER platform is driving over an obstacle.



(a) (b) (c)

Fig. 6: Results of the proof-of-concept experiment regarding robot arm control and collision avoidance cfr. section III-B. Figure 6a shows the generated Octomap of the collision environment found by the RealSense L515 LiDAR camera. Figure 6b depicts the real-life test setup. Figure 6c shows the output of the Python script which calculates a grid to be followed by the robot end effector perpendicular to the segmented plane.

further exploitation of the developed robotic platform in real-life nuclear environments are planned. This will be combined with the benchmarking of the currently used sensor within real-life environments where higher radiation activity could occur. Lastly, a Graphical User Interface will be implemented where the state of the ARCHER platform can be monitored and the operator can switch between tasks that the robot has to execute at any time.

REFERENCES

[1] R. Volk, F. Hübner, T. Hünlich, and F. Schultmann, "The future of nuclear decommissioning – a worldwide market potential study," *Energy policy*, vol. 124, pp. 226–261, 2019.

[2] M. Laraia, *Nuclear Decommissioning: Its History, Development, and Current Status*, ser. Lecture Notes in Energy. Cham: Springer International Publishing AG, 2018, vol. 66.

[3] *Recycling and Reuse of Materials Arising from the Decommissioning of Nuclear Facilities*, ser. Radioactive Waste Management. Paris: OECD Publishing, 2017.

[4] *Occupational Radiation Protection during the Decommissioning of Nuclear Installations*, ser. TECDOC Series. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2021, no. 1954. [Online]. Available: https://www.iaea.org/publications/14858/occupational-radiation-protection-during-the-decommissioning-of-nuclear-installations

[5] J. Kim and B. Tseren, "Occupational alara planning for reactor pressure vessel dismantling at kori unit 1," *International journal of environmental research and public health*, vol. 17, no. 15, pp. 1–11, 2020.

[6] T. Moore, "Robots for nuclear power plants," *International Atomic Energy Agency Bulletin*, vol. 27, no. 3, pp. 31–38, 1985.

[7] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21439

[8] C. Ducros, G. Hauser, N. Mahjoubi, P. Girones, L. Boisset, A. Sorin, E. Jonquet, J. M. Falciola, and A. Benhamou, "RICA: A Tracked Robot for Sampling and Radiological Characterization in the Nuclear Field," *Journal of Field Robotics*, vol. 34, pp. 583–599, 2017.

[9] F. Zhao, Y. Ma, and Y.-L. Sun, "Application and Standardization Trend of Maintenance and Inspection Robot (MIR) in Nuclear Power Station," *DEStech Transactions on Engineering and Technology Research*, 2017.

[10] B. Bird, A. Griffiths, H. Martin, E. Codres, J. Jones, A. Stancu, B. Lennox, S. Watson, and X. Poteau, "A robot to monitor nuclear facilities: Using autonomous radiation-monitoring assistance to reduce risk and cost," *IEEE Robotics and Automation Magazine*, vol. 26, no. 1, pp. 35–43, 2019.

[11] J. C. Chin, D. K. Yau, and N. S. Rao, "Efficient and robust localization of multiple radiation sources in complex environ-

(a) (b)

Fig. 7: Example of the semi-autonomous procedure of scanning a wall with the sensing device perpendicular to the surface as a constraint. Figure 7a shows a single frame in the proof-of-concept scanning procedure. Figure 7b represents the generated heatmap captured during the in-situ proof-of-concept test. The actual source strength was 1.5 MBq coming from an Europium source.

ments," *Proceedings - International Conference on Distributed Computing Systems*, pp. 780–789, 2011.

[12] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4568–4575.

[13] M. S. Lee, D. Shy, W. R. Whittaker, and N. Michael, "Active Range and Bearing-based Radiation Source Localization," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1389–1394, 2018.

[14] F. Mascarich, C. Papachristos, T. Wilson, and K. Alexis, "Distributed radiation field estimation and informative path planning for nuclear environment characterization," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2318–2324.

[15] L. Brabants, M. Simons, D. De Schepper, E. Demeester, and W. Schroeyers, "Minimal detection time for localization of radioactive hot spots in low and elevated background environments using a czt gamma-ray spectrometer," *Nuclear Technology*, vol. 0, no. 0, pp. 1–15, 2022. [Online]. Available: https://doi.org/10.1080/00295450.2022.2073950

[16] M. Nancekievill, A. R. Jones, M. J. Joyce, B. Lennox, S. Watson, J. Katakura, K. Okumura, S. Kamada, M. Katoh, and K. Nishimura, "Development of a radiological characterization submersible rov for use at fukushima daiichi," *IEEE Transactions on Nuclear Science*, vol. 65, no. 9, pp. 2565–2572, 2018.

[17] D. A. Duecker, N. Bauschmann, T. Hansen, E. Kreuzer, and R. Seifried, "Towards micro robot hydrobatics: Vision-based guidance, navigation, and control for agile underwater vehicles in confined environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1819–1826.

[18] A. R. Jones, A. Griffiths, M. J. Joyce, B. Lennox, S. Watson, J.-i. Katakura, K. Okumura, K. Kim, M. Katoh, K. Nishimura, and K.-i. Sawada, "On the design of a remotely-deployed detection system for reactor assessment at fukushima daiichi," in *2016 IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD)*, 2016, pp. 1–4.

[19] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[21] J. Borenstein and L. Feng, "UMBmark: a benchmark test for measuring odometry errors in mobile robots," in *Mobile Robots X*, W. J. Wolfe and C. H. Kenyon, Eds., vol. 2591, International Society for Optics and Photonics. SPIE, 1995, pp. 113 – 124. [Online]. Available: https://doi.org/10.1117/12.228968

[22] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21831

[23] M. Cramer, J. Cramer, D. De Schepper, P. Aerts, K. Kellens, and E. Demeester, "Benchmarking low-cost inertial measurement units for indoor localisation and navigation of agvs," *Procedia CIRP*, vol. 86, pp. 204–209, 2019, 7th CIRP Global Web Conference – Towards shifted production value stream patterns through inference of data, models, and technology (CIRPe 2019). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827120300585

[24] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," 2014.

[25] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com

[26] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[27] J. S. K. Laboratory, "JSK ROS," https://jsk-docs.readthedocs.io/projects/jsk_recognition/en/latest/overview.html, 2015, [Online; accessed 2-September-2021].

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

[29] D. De Schepper, M. Simons, L. Brabants, W. Schroeyers, K. Kellens, and E. Demeester, "Kromek-ROS: A ROS package for interfacing with a Kromek CZT detector," KU Leuven, Tech. Rep., 2021, internal report.

[30] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*, 2017.