Control architecture of a remotely controlled vehicle in extreme CBRNE conditions

Ana Šelek, Demijan Jurić, Anđela Čirjak, Filip Marić, Marija Seder, Ivan Marković, and Ivan Petrović[1],

Abstract— **In this paper, we present a control architecture for a remotely controlled vehicle. It enables fully autonomous navigation of the vehicle while performing GPS-based waypoint or GPS-based patrolling tasks. The addressed tasks are fundamental for resolving CBRNE threats. The control architecture is based on the Quantum geographic information system (QGIS) and the Robotic Operation System (ROS). QGIS is employed for the implementation of the user interface, whereas for the implementation of the vehicle navigation system, ROS is used. We also present a novel solution for the communication between QGIS and ROS. The execution of the waypoint and patrolling tasks is tested in simulation using the Gazebo simulator and experimentally on a Husky A200 mobile robot.**

I. INTRODUCTION

The industrial facilities, such as chemical, petrochemical, pharmaceutical, processing, mining, energy, etc. plants use or produce enormous amounts of very dangerous substances (chemical, radiological and biological) as well as energy and power resources. These facilities as well as associated warehouses and transport systems between them are often taken as targets of the attacks. Also, these facilities are under the risk of different technological accidents that lead to natural disasters. The aforementioned events and the use of weapons of mass destruction create situations that make crisis management difficult or impossible for human teams due to environmental conditions. These conditions correspond to chemical, biological, radiological, nuclear and explosive threats (CBRNE threats) such as open fire and high temperature, explosions and fragmentations, collapsing buildings, danger of improvised explosive devices, explosive mine devices, high intensity electromagnetic (ionizing) radiation, high concentration of dangerous chemical and biological substances, small concentrations of oxygen, conditions of war and terrorist activity in the area of intervention.

Under such conditions, due to the impossible or delayed intervention of the first response teams, accidents cannot be adequately and safely managed on time. Accidents take uncontrolled development and cause enormous material damage, human casualties and disastrous impacts on the environment. Well known examples are the Bhopal gas tragedy, the Seveso industrial accident, the Chernobyl and the Fukushima Daiichi nuclear disasters. These events have a huge political, economic and safety impact on the ongoing positive trend of globalization and the development of a safe society

Unlike human teams, robots and remotely controlled vehicles can easily access dangerous sites, remove existing and potential threats and find and extract victims. Recently, a remotely controlled firefighting vehicle Colossus was used to extinguish the Notre Dame blaze [1]. In 1998 a bulldozer-like robot Pioneer was deployed at Chernobyl [2] to monitor the state of the shut down facility. Also, in [3] authors describes the emergency response to the nuclear accident at the Fukushima Daiichi using mobile robots.

The aim of our project* is to develop a unique remotely controlled vehicle that can sustain the conditions in the extremely hot zone, which humans cannot survive. The vehicle must perform different tasks, such as waypoint, reconnaissance and patrolling, under CBRNE conditions. There are quite many papers published on these topics, proving that the detection and the inspection of the CBRNE threats are important problems. For example, in [5] authors present eight robotic CBRNE incident response tasks. Furthermore, in [4], [6] and [8] authors describe development of CBRNE reconnaissance unmanned ground vehicles for military purposes. They use GPS-based waypoint navigation approach with collision avoidance. In [7] authors describe obstacle avoidance and path planning in an environment under CBRNE terrorist attack. These extreme conditions also pose a problem in maritime environments, and for that reason a system of aerial vehicles was developed for operations in hazardous maritime environments [9].

In this paper we propose the control architecture for our remotely controlled vehicle, which enables semi-autonomous execution of the GPS-based waypoint and patrolling tasks. The control system design is based on the Quantum geographic information system (QGIS), for implementation of the user interface, and the Robotic Operation System (ROS), for implementation of the vehicle navigation system. We also present a novel solution for the communication between QGIS and ROS. The control system architecture and the execution of the waypoint and patrolling tasks are tested in simulation using Gazebo simulator and experimentally on a Husky A200 mobile robot.

II. CONTROL SYSTEM ARCHITECTURE

The architecture of the proposed vehicle control system is depicted in Fig. 1. This control system enables fully autonomous navigation of the vehicle while performing *GPS waypoint* or *GPS patrolling* tasks, which are fundamental tasks for addressing CBRNE threats. GPS waypoint, in terms

*The authors thank DOK-ING company (`www.dok-ing.hr`) for providing technical data of a remotely-controlled vehicle for operation in extreme CBRNE conditions.

[1]University of Zagreb, Faculty of Electrical Engineering and Computing,Laboratory for Autonomous Systems and Mobile Robotics (LAMOR),Zagreb, Croatia {ana.selek, demijan.juric, andela.cirjak, filip.maric, marija.seder, ivan.markovic, ivan.petrovic}@fer.hr

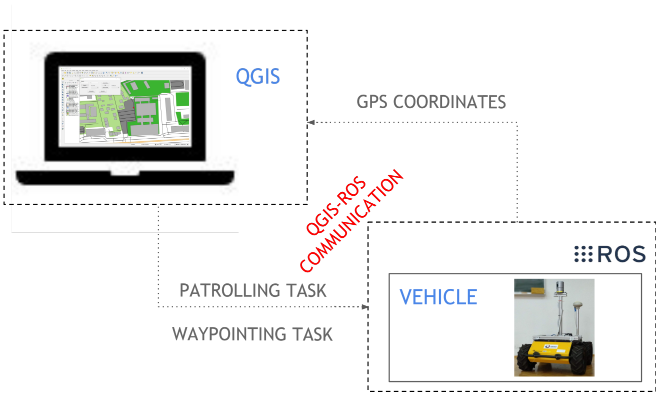

Fig. 1. Control architecture of the autonomous navigation.

Fig. 2. QGIS multi-layer representation.

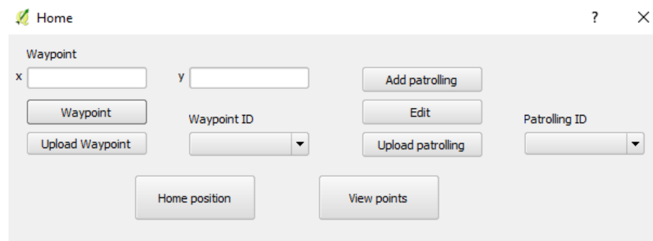

Fig. 3. GUI of QGIS plugin for vehicle's route creation or selection.

of navigation, corresponds to a series of reference points in the physical world. Path planning of the vehicle is confined to specific paths where the coordinates of the goal to be reached are given to the robot known as coordinates. Those specific points are being chosen one by one on a map containing geo-referenced data. It has been shown that robot equipped with only a GPS receiver has obtained good results for waypoint navigation [15]. On the other hand, GPS patrolling focuses on completing a whole sequence of reference points. The successive string of goals is often termed as a route. It is easily comprehended that the waypoint problem presents a subset for the patrolling task. The patrolling task is also often referred to as reconnaissance.

A. QGIS

A geographic information system (GIS) is a computer-based system designed to capture, prepare, store, manipulate, analyze, manage and present geo-referenced data. The GIS technology has wide applicability and can be used for scientific research, resource management, planning, logistics, cartography and road planning. Data represents real objects (such as roads, buildings, waterways, land, etc.) which can be divided into two abstractions: discrete objects (e.g., a house) and continuous fields (e.g., elevations). For both abstractions, there are two methods for storing data: raster and vector method. Raster method stores image data while the vector method uses geometry object like point, line or polygon for the representation of data. To represent data in the form of the map of some environment, it is allowed to split it into layers, for example, one layer for buildings and another for roads [11]. See Fig. 2 for an example of multi-layer map representation.

QGIS is a free open-source geographic information system that works on Windows, Linux, Mac OS, and Android platforms. Extended functionality of QGIS are plugins which can be written in Python or C++. To fulfill the requirements for remote control and autonomous navigation, we developed a QGIS plugin. For the development of this plugin, QGIS, Qt4, and PyQt4 were used. The user-friendly interface of the plugin was made in the Qt4 designer. The translation of Graphical User Interface (GUI) created with Qt toolkit was made with PyQt4 [12].

The main task of the developed QGIS plugin is to create a desired route that can be assigned to the robot for autonomous navigation. The route can be created as a sequence of waypoints, i.e., to define a GPS patrolling task, or as a selection of a single waypoint, i.e., to define a GPS waypointing task. The coordinates are then saved through its ID into its corresponding layers. Afterwards, the coordinate conversion is performed, which consists of extracting the latitude and longitude coordinates and converting them to the Universal Transverse Mercator (UTM) coordinate system. Finaly, UTM coordinates are forwarded to the robot as goals. The robot accepts these goals within ROS layer, for which QGIS–ROS communication, and type of messages are developed (confer Section II-B).

The developed QGIS plugin is used through its GUI (Fig. 3), which contains buttons for defining various tasks. The **Home position** button shows the current position of the vehicle in the map of QGIS and saves it into the *home layer* as coordinates in a latitude-longitude coordinate system. If the vehicle is moving, its current path can be visualized in QGIS with the **View points** button. The driven path is continuously saved as a discrete set of points in the *point layer* in latitude-longitude coordinate frame, where two consecutive points keep distance for a predefined value (1 meter in our implementation).

The **Waypoint** button enables the user to assign a goal point. Afterwards, the user needs to select the point on the map of QGIS visualization interface. The goal point is saved in the *waypoint layer* with its *Waypoint ID*. The coordinates are in the latitude-longitude coordinate system and must be converted to the UTM coordinate system. UTM coordinate system is a geographical latitude-longitude system that is

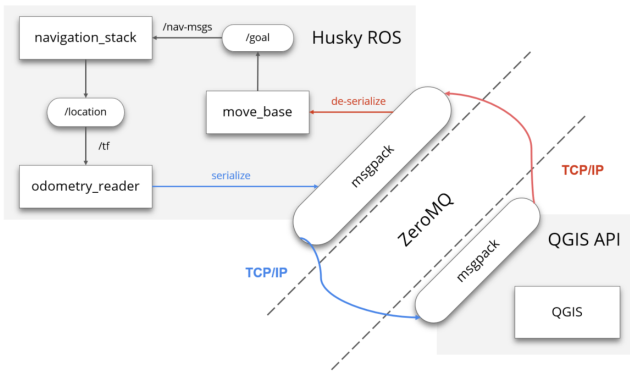

Fig. 4. Outline of the communication infrastructure, which transmits user's commands and data from the vehicle's sensor array.

expressed in a two-dimensional projection of the surface of the earth. The earth map is divided into 60 zones, where each zone is separated by 6 degrees in longitude from East to West. The locations are expressed like Easting for the x coordinate and Northing for the y coordinate [13]. Within the QGIS plugin, we implemented the calculation of the right zone for the conversion to UTM. The converted goal points in UTM are then forwarded to the vehicle with the **Upload Waypoint** button.

The **Add patrolling** button assigns route to the vehicle. Routes are saved as a series of points with a *Patrolling ID* into a new *patrolling layer* and can be further edited by user with the **Edit** button or sent to a vehicle with the **Upload patrolling** button. Additionally, driven path saved in the *point layer* can also be edited. Therefore, patrolling along the previously saved routes already passed by the vehicle, while under remote control, can also be used as an option.

B. QGIS and ROS communication

A key requirement for successful interfacing and teleoperation of the robot is an efficient and robust communication infrastructure. Our infrastructure connects the user-facing QGIS software to the underlying low-level ROS network graph using the lean ZeroMQ networking library [16]. That way, the information collected by the robot may be displayed to the user, while user inputs are sent to the ROS infrastructure. As demands may vary for specific classes of tasks, our goal is to make this infrastructure modular and content-agnostic.

As can be seen in Fig. 4, new goals are defined by the user as locations in the form of GPS coordinates. Once defined, the goals are selected in the QGIS environment and passed to the vehicle as motion commands. Goal data is then gathered using the QGIS API, serialized into messages using the *msgpack* library, and sent to the vehicle computer. Reciprocally, observations gathered by the vehicle's sensors are passed to the user's system. Message passing is performed using the modular, platform-agnostic *ZeroMQ* library, which uses the TCP/IP protocol. Once the messages are received by the vehicle computer, they are unwrapped into standard ROS messages and passed within the navigation infrastructure using the ROS node graph.

Motion goals and associated task IDs are sent from the central QGIS level to the local ROS level. Goals are sent as an array of 3D coordinates in the map frame, while the task ID is sent as an integer-character pair describing the ordering and grouping of individual waypoints, respectively. When the robot receives the first goal position it can not receive a new goal position until it has reached the first goal. This is controlled by the ROS *action server* within the navigation stack.

Similarly, data describing the current location and status is transmitted back to the QGIS level from ROS. This includes the current task ID, an error flag indicating expected operation, the current location of the robot in the map coordinate frame, and current battery status. The modular paradigm of the communication protocol allows for simple extension or modification of these messages to include additional task-specific information. It is also worth noting that the rate at which messages are received depends on the robot's travelled distance, where a lower transmission rate corresponding to a greater distance of the robot from the QGIS device. Consequently, new messages are transmitted from the robot every time a predefined distance is traveled from the last logged position.

C. Navigation

The localization pipeline for the robot integrates wheel odometry, IMU and GPS information, producing a global estimate of the robot's location and orientation. In order to achieve the above goal, while maintaining a high degree of robustness and extendability, we used the well documented *robot_localization* ROS package. Local planning is performed in the odometry frame, where the robot's position and orientation is updated at a high frequency by an extended Kalman filter (EKF), which fuses the IMU and wheel odometry data. The GPS information arrives at a lower frequency, and is used for global position monitoring and waypointing through the QGIS interface. This way, sudden changes in the position estimate due to the introduction of new GPS data are avoided in the local planning process.

We used a receding horizon control (RHC) navigation developed within our research group [14]. RHC navigation is an on-line optimization algorithm that predicts system outputs based on its current states and mathematical model of the mobile robot differential drive. RHC objective function utilizes a local-minima-free navigation function to measure the cost-to-goal over the robot's trajectory. The navigation function is derived from the path-search algorithm over a discretized 2D search space, which is updated as the environment changes. The RHC navigation algorithm is used because it produces fast motion to the goal with low computational costs. It is adapted to be used inside the ROS navigation stack with the *action server* for controlling the current robot's actions.

III. SIMULATION RESULTS

The previously described GPS waypoint and GPS patrolling are tested in Gazebo simulator, which simulates the map and the robot Husky. The grid cells in QGIS had a size of 1×1 m^2, where the driven robot path is presented as described in II-A.

A. GPS waypointing

Figure 5 (left) shows the route obtained by autonomous navigation presented at the QGIS (central) level as a sequence of points from home position (marked as pentagon) to the given waypoints (marked as a star). After reaching each goal the new waypoint execution is started. Figure 5 (right) shows the ROS (local) level with received waypoints and a robot path in local frame. Each waypoint has the corresponding index starting from 1. It can be seen that the same waypoint is marked with two numbers (2 and 3) due to the intentionally sending of the same waypoint twice, which did not provide a fault or stop of the waypointing task. Comparing left and right figures in Fig. 5, it can be seen that home position, waypoints, and paths driven by the robot in the simulator are almost identical. This also indicates that QGIS-ROS communication works well and the data is reliable for surveillance of autonomous robot navigation.

B. GPS patrolling

Figure 6 (left) shows the patrolling test at the QGIS level and a route obtained by autonomous navigation from home position (marked as pentagon) through every point on the given route (marked as dashed lines). Every change of direction on the given route is saved as a goal point into the array and forwarded to the robot. In this example, the robot visited 6 goals and stopped when it reached the last goal position. Figure 6 (right) shows the traveled trajectory of the robot at the ROS level. All six patrolling points are received as goal points, which are labeled with numbers from 1 to 6. Comparing and analyzing left and right subfigures in Fig. 6, it can be seen that driven trajectories have similar shape, and the robot was able to drive the given patrolling route.

IV. EXPERIMENTAL RESULTS

The Husky robot was placed in front of the building of University of Zagreb, Faculty of Electrical Engineering and Computing, where it received the GPS signal, where the Home position can be seen as point in the middle of Fig. 2. The robot is equipped with an Intel NUC mini PC running ROS. The mini PC is directly wired to the sensors used in the experiment. The sensors include the encoders providing the robot's wheel velocities, Xsens MTi-G-710 IMU offering velocity and acceleration measurements, and the SMART-V1G Novatel integrated L1 GPS + GLONASS receiver and antenna. The grid cells in QGIS had a size of 3×3 m^2, where the driven robot path is presented as described in II-A.

A. GPS waypoint

Figure 7 shows the GPS waypointing test using the Husky A200 mobile robot in a real environment. Six waypoints (stars) were assigned in QGIS (Fig. 7-left) and were received by the robot that traversed these goals one by one starting from the Home position (pentagon). Figure 7 (right) shows the same waypoints and driven path in the local robot coordinate frame. All received waypoints are indexed with labels from 1 to 6. The same waypoint was intentionally sent twice (labeled with 2 and 3), which did not produce the failure of the GPS waypointing test. Comparing these two figurses in Fig. 7, it can be seen that Home position, goals, and paths driven by the Husky A200 mobile robot are quite similar. However, there exist some discrepancies in the trajectory due to the much higher noise of the GPS sensor in reality. The largest difference is near the Home position. The successful test also indicates that QGIS-ROS communication works well and the data is reliable for surveillance of autonomous robot navigation without significant errors in path visualization at the QGIS level.

B. GPS patrolling

Figure 8 shows results of the GPS patrolling test of autonomous functionality. In Fig. 8 (left) the route is assigned as dashed line, which represents the path which robot needs to autonomously traverse. Every change of direction of the line is saved as a goal point into the array and forwarded to the robot. In this example, the robot visited 6 goals and stopped when it reached the last goal position marked with number 6. It can be seen that the robot's trajectory in the real experiment has much larger discrepancy from the desired route than in simulation. This was to be expected due to the noise of the real GPS sensor and the significant error of the robot's orientation which is estimated from odometry and IMU data. Figure 8 (right) shows the traveled path of the Husky A200 mobile robot in a real environment in the local robot coordinate frame. Comparing the left and right figures in Fig. 8, it can be seen that visualized path in QGIS is quite similar to the local robot path, which proves that it is reliable for surveillance of autonomous robot navigation from a remote computer.

V. CONCLUSION

Our work suggests an efficient and robust communication method for autonomous mobile robots and vehicles. The two-way ROS-QGIS link proves to provide a reliable link for navigation via a remote station i.e. desktop computer. The sensor array presented can be easily expanded to include camera data logging and LiDAR scans. The established system stands as a stepping stone for further research especially in the field of autonomous recovery missions. Additionally, the next step would contain environment mapping by LiDAR scans to provide fully autonomous regress operation and even stronger robustness and safety. The modularity of our solutions lends to immaculate portability to other platforms or vehicles.

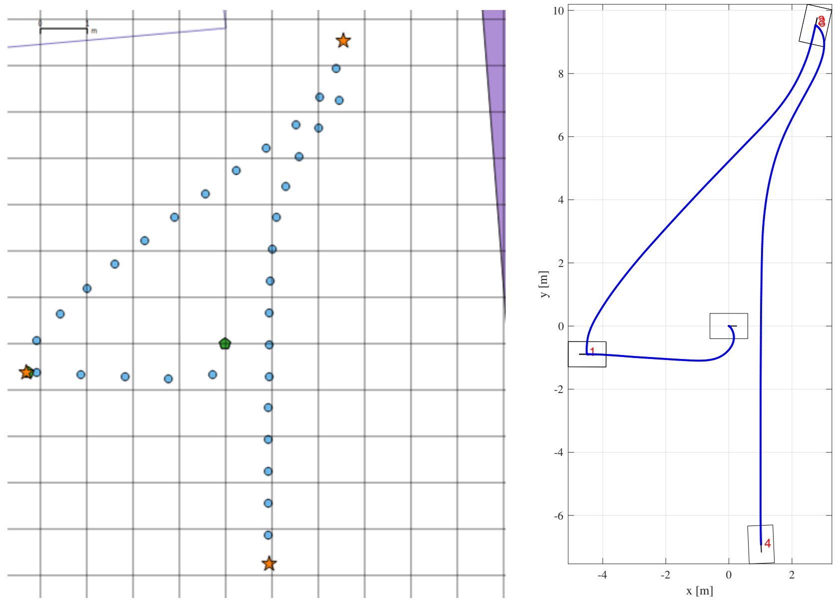

Fig. 5. The GPS waypoint test of autonomous functionality simulated in Gazebo. Left: QGIS visualization of waypoints assignment (stars) and the traveled trajectory (points). Right: the traveled trajectory by the simulated robot Husky in the local coordinate frame.

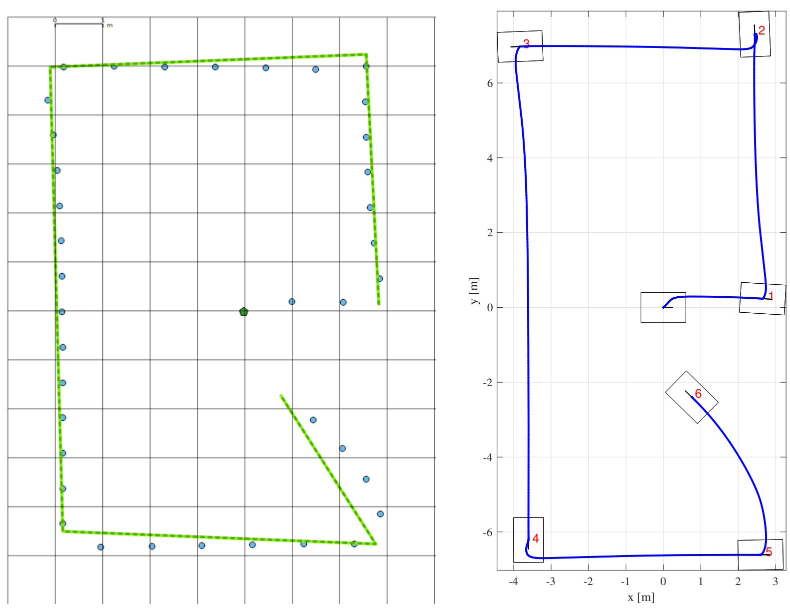

Fig. 6. The GPS patrolling test of autonomous functionality simulated in Gazebo. Left: QGIS visualization of the assigned route (dashed line), and the traveled trajectory (points). Right: the traveled trajectory by the simulated robot Husky A200 in the local coordinate frame.

ACKNOWLEDGMENT

This work has been supported by the European Regional Development Fund under the grant KK.01.2.1.01.0045 - Development of a remotely-controlled vehicle for operation in extreme CBRNE conditions (DUV-NRKBE).

REFERENCES

[1] L. Peskoe-Yang, "Paris Firefighters Used This Remote-Controlled Robot to Extinguish the Notre Dame Blaze", IEEE Spectrum: Technology, Engineering, and Science News, 2019. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/industrial-robots/colossus-the-firefighting-robot-that-helped-save-notre-dame. [Accessed: 30-Apr-2019]

[2] J. Abouaf, "Trial by fire: teleoperated robot targets Chernobyl", IEEE Computer Graphics and Applications, vol. 18, no. 4, pp. 10-14, 1998.

[3] K. Nagatan et al., "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.

[4] F. E. Schneider, J. Welle, D. Wildermuth and M. Ducke, "Unmanned multi-robot CBRNE reconnaissance with mobile manipulation System description and technical validation," Proceedings of the 13th International Carpathian Control Conference (ICCC), 2012., pp. 637-642

[5] C. Humphrey and J. Adams, "Robotic Tasks for Chemical, Biological, Radiological, Nuclear and Explosive Incident Response", Advanced Robotics, vol. 23, no. 9, pp. 1217-1232, 2009.

[6] F. E. Schneider and D. Wildermuth, "An autonomous unmanned vehicle for CBRNE reconnaissance," 2011 12th International Carpathian Control Conference (ICCC), 2011., pp. 347-352

[7] T. Ohki, K. Nagatani and K. Yoshida, "Local Path Planner for Mobile

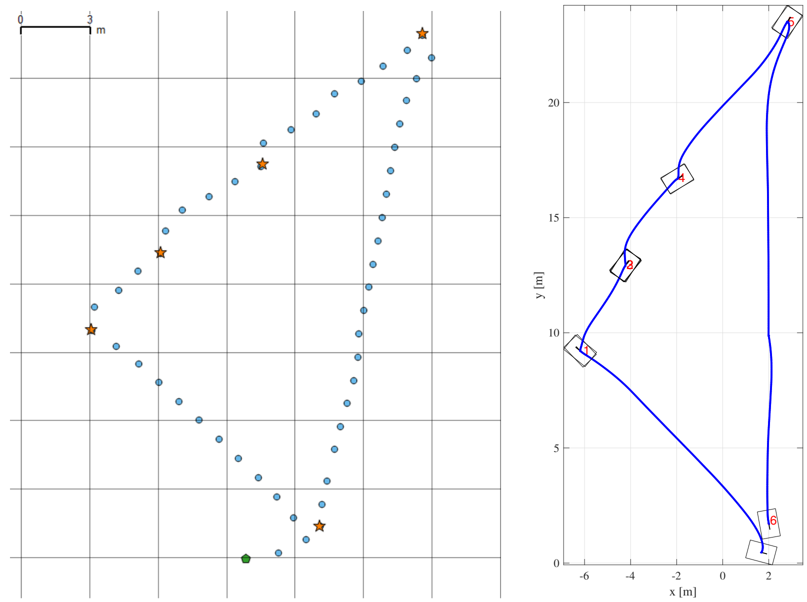

Fig. 7. The GPS waypointing test of autonomous functionality in a real environment using the robot Husky A200. Left: QGIS visualization of waypoint assignments (stars) and the traveled trajectory (points). Right: the traveled trajectory by the robot Husky A200 in the local coordinate frame.

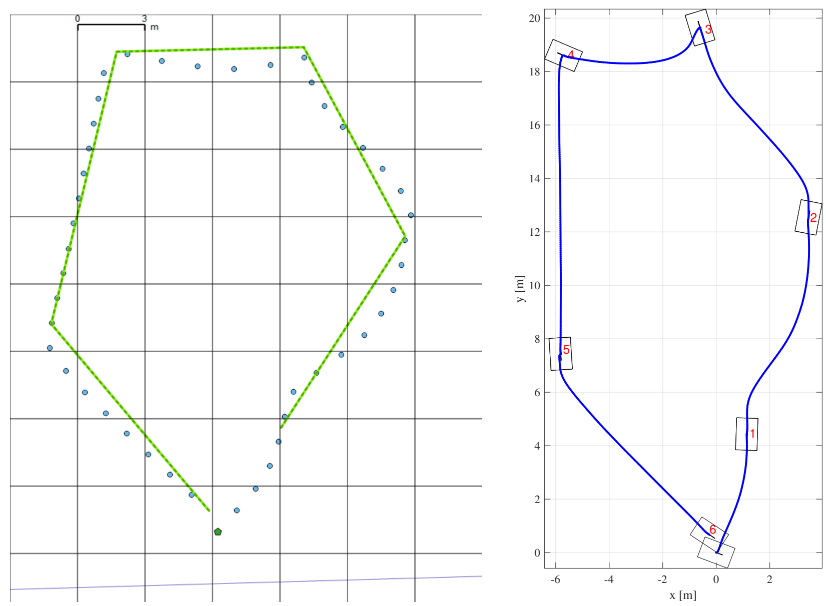

Fig. 8. The GPS patrolling test of autonomous functionality in a real environment using the robot Husky A200. Left: QGIS visualization of the assigned route (dashed line), and the traveled trajectory (points). Right: the traveled trajectory by the robot Husky in the local coordinate frame.

Robot in Dynamic Environment based on Distance Time Transform Method", Advanced Robotics, vol. 26, no. 14, pp. 1623-1647, 2012.

[8] F. Schneider, T. Röhling, B. Brüggemann and D. Wildermuth, "CBRNE reconnaissance with an unmanned vehicle - A semi-autonomous approach -", IFAC Proceedings Volumes, vol. 43, no. 23, pp. 122-127, 2010.

[9] M. M. Marques et al., "GammaEx project: A solution for CBRN remote sensing using unmanned aerial vehicles in maritime environments," OCEANS 2017 - Anchorage, 2017., pp. 1-6

[10] R. Guzman, R. Navarro, J. Ferre and M. Moreno, "RESCUER: Development of a Modular Chemical, Biological, Radiological, and Nuclear Robot for Intervention, Sampling, and Situation Awareness*", Journal of Field Robotics, vol. 33, no. 7, pp. 931-945, 2015.

[11] T. Sutton, O. Dassau, M. Sutton. A Gentle Introduction to GI, Spatial Planning and Information, Department of Land Affairs, Eastern Cape, 2009

[12] O. Huisman,R.A. De By, Principles of Geographical Information Systems,ITC, Enschede, Netherlands, 2009.

[13] M. Baisantry, V.K. Saxena, Removal of ladder effects by smoothening the integer format elevation layer via development of a QGIS plugin. 2016 International Conference on Computing, Communication and Automation (ICCCA), pp. 1054-1058. April 2016.

[14] M. Seder, M. Baotić, and I. Petrović, Receding horizon control for convergent navigation of a differential drive mobile robot, IEEE Trans. Control Syst. Technol., vol. 25, no. 2, pp. 653-660, March 2017

[15] S. Shair, lH. Chandler, Y.l Gonzalez-Villela, R.M. Parkin, and M.R. Jackson, The Use of Aerial Images and GPS for Mobile Robot Waypoint Navigation, IEEEIASME Trans. on Mechatronics, vol. 13, no. 6, pp. 692-699, December 2008.

[16] Hintjens, P. (2013). ZeroMQ: messaging for many applications. " O'Reilly Media, Inc.".