JOGSC

Journal of Global Science ISSN: 2453-756X (online) http://www.jogsc.com

Navigation solution of autonomous robot system in the narrow area

Ing. Branislav Rácek ,
Department of strategic project department
Railway Company Slovakia
Bratislava, Slovakia
racek.branislav@slovakrail.sk

Ing. Róbert Hudák ,
Department of strategic project department
Railway Company Slovakia
Bratislava, Slovakia
hudak.robert@slovakrail.sk

Ing. Peter Tomašovič ,
Department of strategic project department
Railway Company Slovakia
Bratislava, Slovakia
tomasovic.peter@slovakrail.sk

Ing. David Hudák ,
Technical University of Košice
Košice, Slovakia
davidhudak.dh@gmail.com

doc. Ing. Michal Hovanec, PhD, Ing.Paed. IGIP
Department of strategic project department
Railway Company Slovakia
Bratislava, Slovakia
hovanec.michal@slovakrail.sk

Abstract
Scientific laboratories have been working hard to study the spread and behaviour of agents in the native environment during the COVID-19 pandemic. Research has shown that the virus survives on surfaces for up to several days. This reason led scientists and engineers to cooperation and interdisciplinary cooperation of several fields. Their goal was to reduce the exposure time of the worker to the contaminated space or surface. With the help of modern technologies, it was possible to build and manufacture disinfection robots. These machines reduced the risk exposure time to zero, as they were fully autonomous.

Key words
navigation, robot operating system, point – to – point

Acknowledgement
This work was supported by the Slovak Research and Development Agency under the contract number OPIIVA/DP/2020/9.4-01 within" Safe mobility in the framework of the pandemic caused by the disease COVID-19: safe operation and use of means of passenger rail transport" project implemented under contract number 313011ATQ2.

1. Introduction

Robotics is an interdisciplinary field and, together with automation, are key areas that are rapidly developing and have great potential in many industries, such as manufacturing, logistics, research and development, and medicine. In recent years, robot development has moved from research to the commercial sector and is becoming increasingly important for the future of jobs and the economy [1].
Programmatic implementation in robotics using the ROS framework is increasingly being used. ROS or the full name robot operating system is a set of software libraries, packages and tools that are provided under an open-source license and used by developers, companies, and robotics enthusiasts all over the world with the purpose of creating their own autonomous robotic applications. Currently, it is the most widespread open-source middleware in the field of robotics. ROS architecture is used as a tool. Its basic components and the method of implementing a robotic solution for autonomous disinfection, specifically of train spaces. The technical teams met with various options for solving implementation procedures and thus choosing a suitable operational architecture. Two architectures are known, ROS1 and ROS2. their similarity allows for fluent translation of the language using simple "bridges".
All over the world, various large companies create their own internal software packages covering the necessary functionalities, communication, navigation, and all components related to robotics and their autonomy. Robotic middleware provides a framework for running and managing complex robotic systems from a single interface [2]. ROS middleware hides the complexity of low-level communication, provides sensor heterogeneity, and creates room for the reuse of robotic software infrastructure across multiple research efforts. This fact globally reduces production costs, for example, also in research activities. The availability and relatively large community in ROS facilitate the development itself and create a wider space for the use of acquired knowledge compared to other technologies [3].
ROS provides a number of tools and libraries for robot programming, hardware-software communication, and resource management. We will deal with the implementation of robotics in the ROS environment. It will present the basic principles of development in ROS and the method of implementing a specific project of a robotic solution for the autonomous disinfection of train spaces.

2. Basic theory

Among the currently most used software for the development of robotics in ROS are RVIz and Gazebo, which were also used in the development of a robotic solution for the autonomous disinfection of railway vehicle spaces [4]. Both software are similar to each other but each has a different purpose. Gazebo shows the reality of the simulated world and RVIz shows the perception of the robot. Often, both software were used at the same time when the necessary hardware was not available, in the sense that a wagon environment was created in the Gazebo, which was scanned by 3D scanners and then imported as a 3D model into the Gazebo environment. The individual sensors and navigation were tested in the RVIz environment. RVIz, in addition to showing the perception of individual peripherals or "senses" of the robot, also shows us the global and local path planner and offers the possibility of defining a colour spectrum for each necessary part, which was very useful in the project when debugging, for example, a bad movement trajectory.

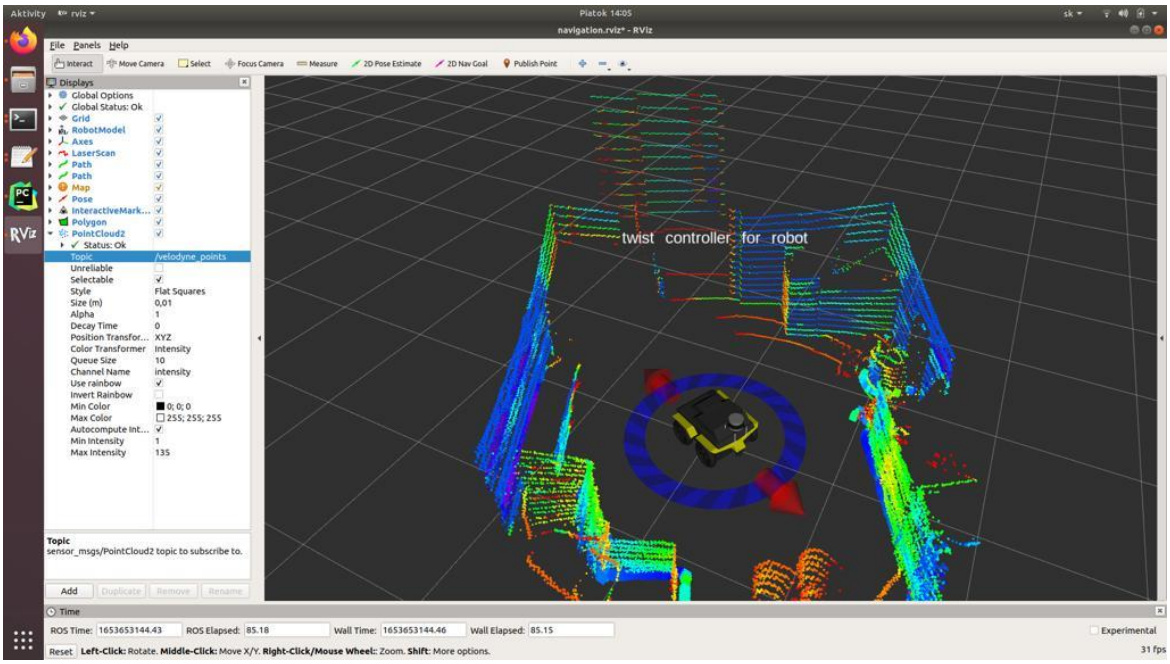

Fig. 1 Pracovné zobrazenie v RVIz

2.1 Global and local planner

The ROS global and local planners are components within the ROS navigation stack that are responsible for path planning for a robotic unit [5]. While the global scheduler generates a long-term, high-level plan for the robot to follow, the local scheduler generates a detailed short-term plan that the robot can follow while executing the global plan.

The global robot planner built on the ROS framework generates a path by searching the environment to find a collision-free path from the robot's current location to its destination. It generates a high-level plan that the robot can follow, taking into account the limitations of the robot and the environment in which it operates.

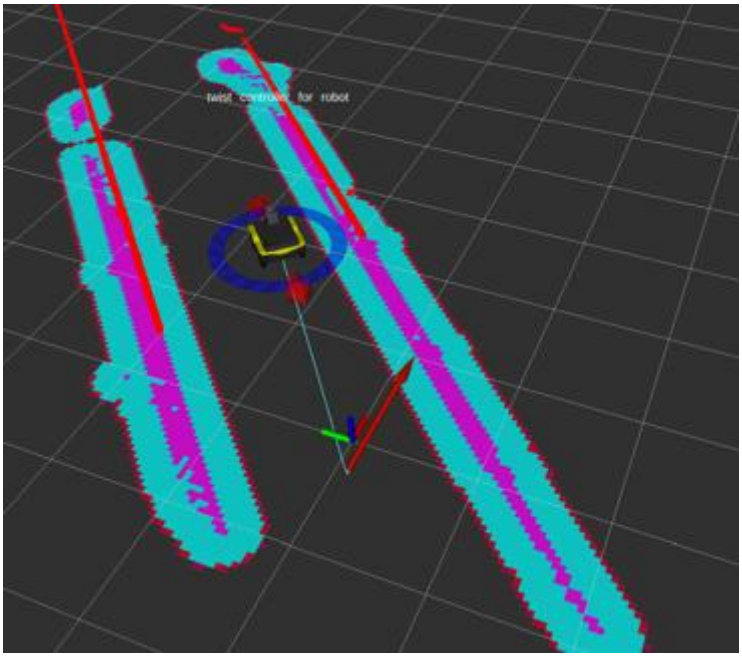

Fig. 2 local planner view

The local planner, on the other hand, is responsible for creating a more detailed plan that the robot can follow while executing the global plan. Thus, the local scheduler takes the high-level plan from the global scheduler and generates a set of low-level control commands for the robot to follow. The local planner ensures that the robot's movements are safe, smooth, and efficient, considering factors such as the robot's speed and acceleration limits, obstacles in the environment, and dynamic constraints such as the robot's orientation. Together, the global and local planners form the core of the ROS navigation system and provide a powerful tool for controlling and coordinating robot movements in complex environments.

3. Methodology

The Passage navigator node is a set of methods used to navigate the robot when passing between cars. It is an extension of the Straight navigator. The input is a cloud of points (PointCloud2) and two straight lines that adapt to the space in which the robot moves. We get a cloud of points by listening to the topic /merged points, which is broadcast at a frequency of 10 Hz. It was created by combining the scans of the front lower 2D lidar (180°), the rear lower 2D lidar (180°), and the front upper 2D RPlidar (360°) [6]. Straight lines approximating the space can be obtained by listening to the topic /lines, which is broadcast at a frequency of +-10 Hz.

The output of the navigator is the message /jackal_cmd_vel which controls the movement of the robot (translation and rotation), with a frequency of +-10 Hz.The navigator starts at the last stop of the Seat navigator.The navigator ends after the robot enters the carriage (seat area) and provides information about the entrance side (on which side the door is) (3 types):

- Middle
- Right
- Left

If the Jackal enters the carriage in its middle, the Seat navigator follows (eg an Ampeer type carriage). The side seat navigator follows at the entrance to its left or right part. It happens in wagons of the Bempeer or Bdmpeer type, where it is necessary to make a handle to catch the first stop of the seats. The side seat navigator is an extension of the Seat navigator enriched with this handle.
The following image shows the Passage navigator and the passage of Ampeer-Bempeer type wagons. Jackal enters the right part of the carriage (seat area), the Passage navigator is switched off and the Side seat navigator is switched on. The handle follows and continues with the Seat navigator.

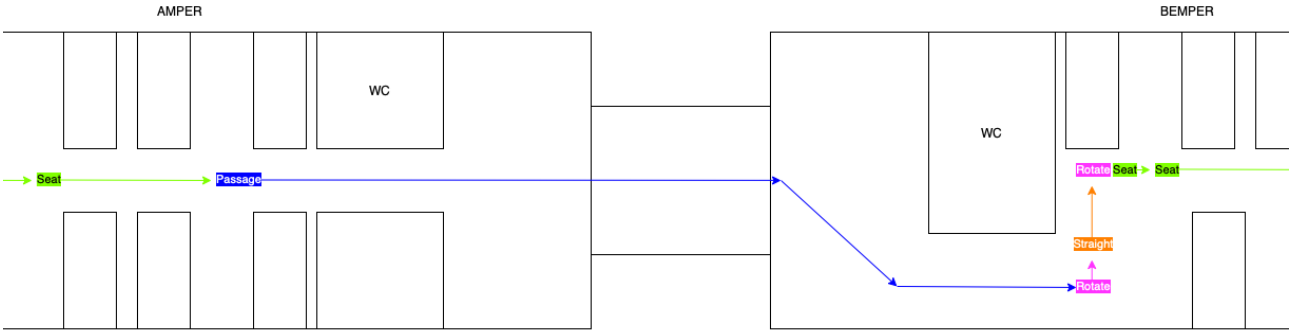

Fig. 3

The next picture shows the Passage navigator and the passage of Amper-Amper carriages. Jackal enters the middle part of the carriage, the Passage navigator is switched off and the Seat navigator is switched on.

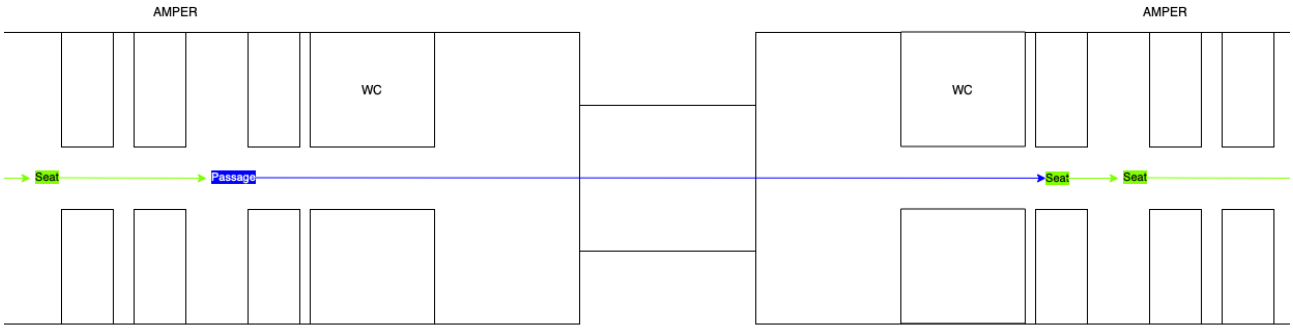

Fig. 4

The side of the entrance (centre, left, right) should be understood as something like the detection of the type of wagon. Based on this, we would be able to tell in which wagon the Jackal is located. The detection of the entrance to the carriage (seat area) is performed by the detection entrance service, which will be described below. If this service receives the information that the Jackal has entered the seating area, the Seat navigator or Side seat navigator is activated.

3.1. Detection entrance service

This service is used to detect entry into the seating area of the wagon. The service provides information on when the robot entered the seating part of the wagon and in which part of it, respectively. where is the door (left, right, centre). It is called continuously for the entire duration of the navigator's Passage, with a certain frequency, e.g. 2 or 5 Hz.
Which part of the wagon Jackal entered is important so that the robot knows which navigator it needs to start next, or also which type of wagon it is in.

3.2. Principle

How to find out when the robot entered the seating area of the carriage? The Passage navigator is started at the last stop of the Seat navigator and lasts until the Seat navigator is started again in the next car. The easiest way to detect this is with the front bottom lidar by detecting the width of the car. At the moment when the robot enters the seating area, the carriage is wide, e.g. more than 2 meters. But that is not enough for us, because in the picture below we can see that during the period when the Passage navigator is turned on, the carriage is this wide in three places (blue points 1, 2 and 3).

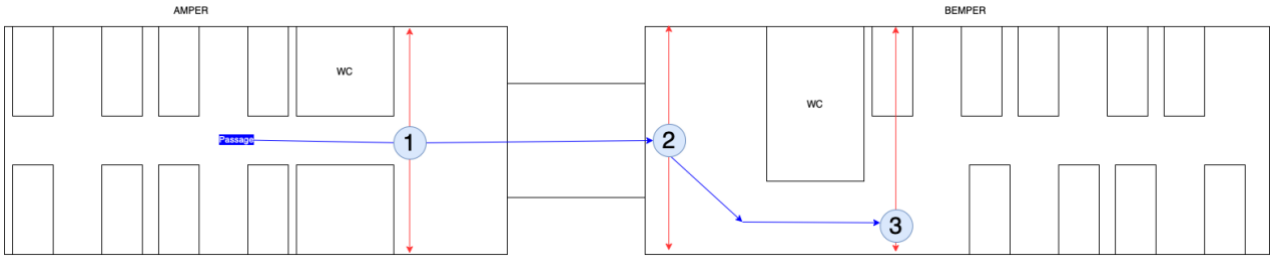

Fig. 5

So, we can take the front lower lidar again to help. It can also be used to detect distant points that the robot is able to see. When he enters the part of the train we are looking for (point 3), we will see points that are distant, e.g. even more than 10 meters. In the other cases (points 1 and 2), the first assumption will be fulfilled (car width greater than 2 meters), but the furthest point that the robot will see will be only a few meters (less than 10 m).

These two assumptions would be enough for us to combine the wagons in the upper picture. But what happens if the robot moves from an Ampeer type car to the same car? In the following picture, we can see that in such a case, the robot has a view from the first wagon (from point 1 and 2) to the second, i.e. both conditions would be fulfilled, which means that this solution is not enough. Of all three points, Jackal would detect points far enough away to satisfy the second assumption.

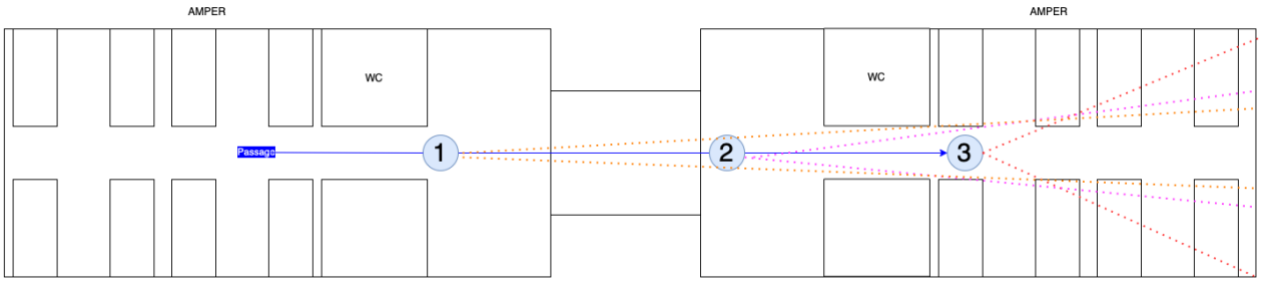

Fig. 6

But now notice the marked angle described by the distant points from the second assumption. When the Jackal enters the part of the wagon we are looking for (point 3), the angle will be larger than in points 1 and 2. At the moments in points 1 and 2, this angle will be smaller. According to this, we can now clearly distinguish whether the robot is in the seating part of the car (point 3) or not.

Next, we will describe how the detection itself works.

3.3. Detection

The first image captures the movement of the robot in the Ampeer type carriage. This is the simplest type of carriage as far as the Passage navigator is concerned. After passing from carriage x to the Ampeer carriage, it is enough for the robot to go straight and simply arrive at the place where the Passage navigator should be switched off and the Seat navigator switched on.
The Detection entrance service function can be divided into four states. In the image, they are interpreted by arrows (green, orange, red) and the forward lidar scan (purple lines).

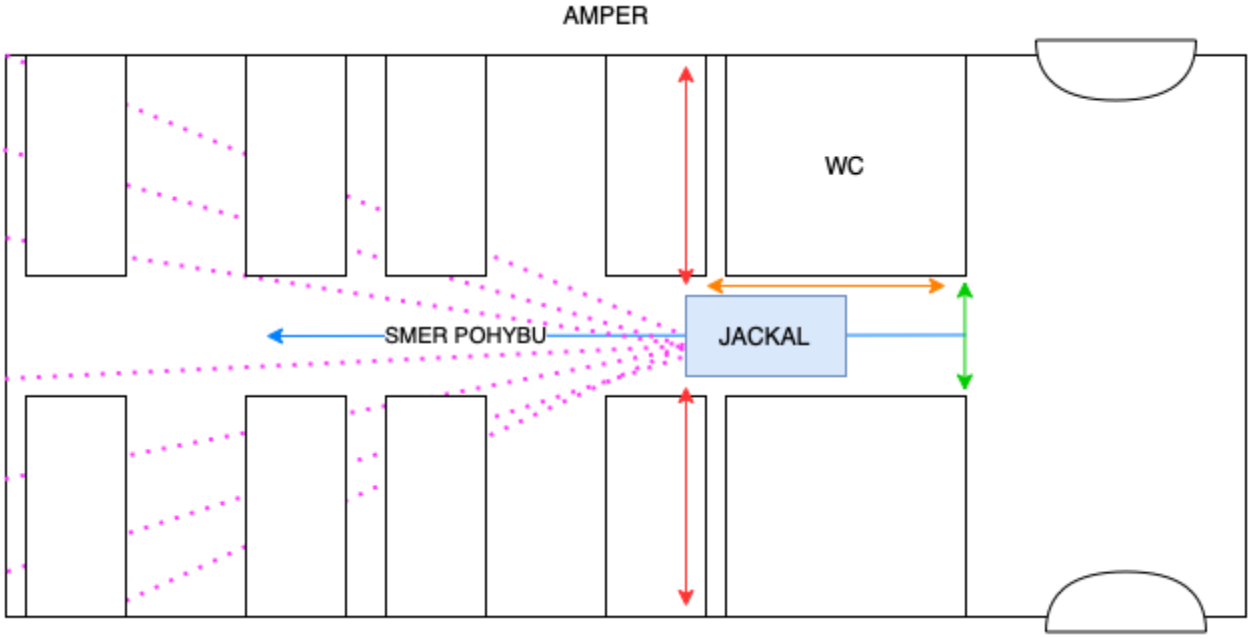

Fig. 7

The first condition that must be met to be able to detect the entrance to the seating part of the car is that the robot must pass through a narrow aisle (which is in every car, therefore it is a suitable identifier) with a defined width (green arrow) for a defined length (orange arrow). In other words, the detection of entry into the seating part of the car is not triggered until the robot passes through a narrow passage with a defined width and length. This is to avoid intrusive/accidental detections of entry into the carriage.
The width can be set to values smaller than 1-0.5 m. The length can also be set to similar values.
If the first condition is met, the second condition (described in the previous subsection - Principle) is awaited, which is the detection of the seat part of the car, in other words, a wide space with a width of more than 2 meters is searched for (red arrows in the picture).
This is the position in which we see the Jackal in the picture.
The third and last condition before switching the navigator is the detection of scanned distant points with a certain width (angle), which is also explained in the previous subsection - Principle.
The robot already knows that it has entered the seating part of the car, and now starts the Seat navigator, which further takes control over it.
In the upper picture, we can see that according to the red arrows, we can clearly say that the robot entered the wagon in its central part. So the Seat navigator follows and the robot continues in a straight line.
In the picture below, according to the red arrows, you can see that the robot is in the left part of the wagon. This is followed by the Side seat navigator and the knob to the right, which will take you to the first stop.

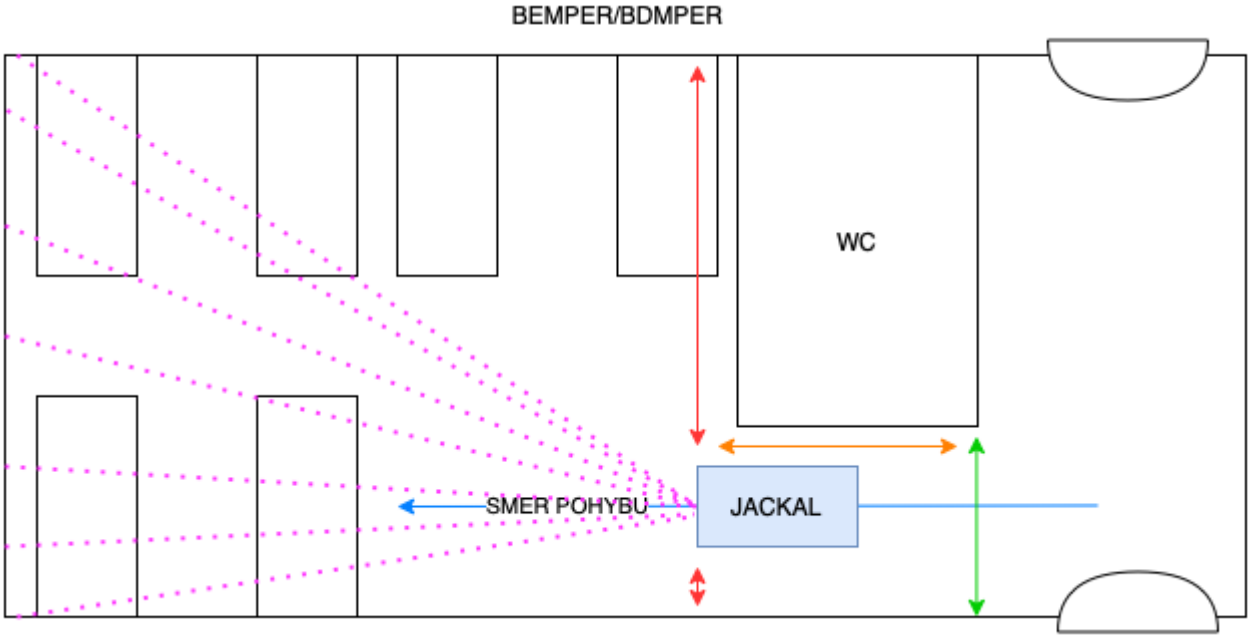

Fig. 8

In the next picture, according to the red arrows, you can see that the robot is in the left part of the wagon. But it is a wrong statement. However, we cannot complain, because thanks to this, we can say that Jackal is in a BMZ-type wagon, and can continue to move in a straight line, not with the Side seat navigator, but with the Seat navigator, just like it was with the Ampeer-type wagon.

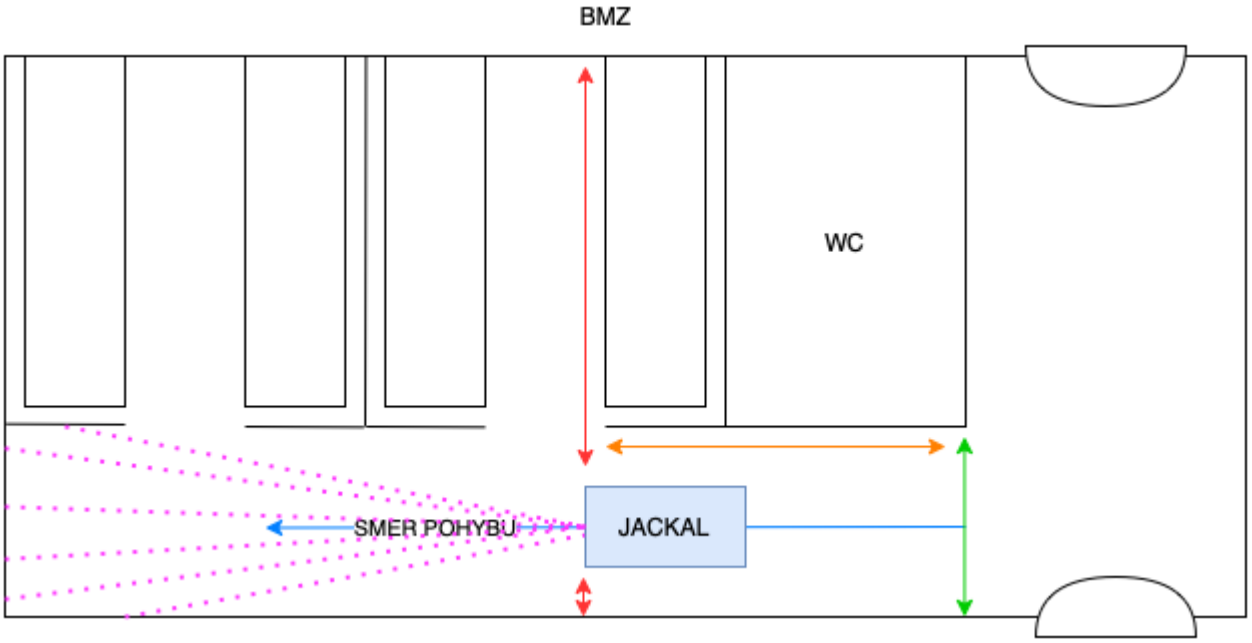

Fig. 9

How is this possible when the angle of distant points is small? So the second assumption is not fulfilled? The second assumption is based on the fact that when passing from an Ampeer type wagon to the same Ampeer type wagon, we must find distant points that have a sufficient angle.

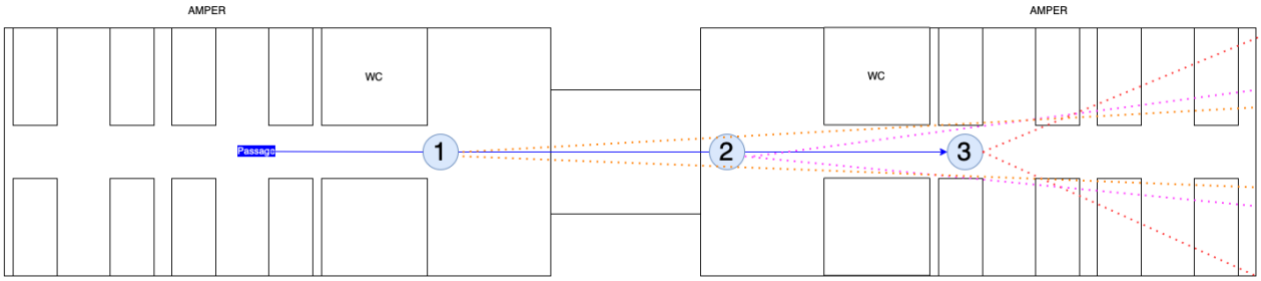

Fig. 10

But we can bypass this condition and say that if the service evaluates that the robot entered either the right or left part of the wagon (it did not enter through the middle), we do not take into account the minimum angle that must be described by distant points. We just need to find some distant points. This is the only exception that applies to the detection of entry into the car.

4. Discussion

Using a map or a navigation algorithm is not always necessary to create a map for autonomous movement and navigation. The space of the wagon is mostly flat and significantly narrow. The current Jackal UGV robotic chassis, which uses the above-mentioned navigation algorithms, and has a graphical user interface ARK, or the full name Autonomy Research Kit, which could not cope with such narrow spaces. When solving the problem of a narrow space through ARK, we tried to reduce the width of the robot in software, but also to change the configuration file in general. As a result, the current navigation system based on SLAM and map navigation, built on ROS1 navigation algorithms, is incapable of autonomous navigation in the narrow spaces of the wagon.

5. conclusion

The robotic solution for disinfecting means of transport has proven to be effective over time corona, when the effort was to ensure the safety of passengers and operational workers services such as cleaning and disinfection. As is known, the means of transport is articulated and the narrow spaces through which the robotic unit must be transported, and therefore sophisticated a solution that will provide stability, accuracy and consistent fulfilment of the task of disinfecting not only surfaces but also the atmosphere. this is ensured by point-finding
transformations.
It is important to orientate and determine the exit points for the robotic chassis, such as the entrance to the space means of transport.

Zoznam bibliografických odkazov

1. Hanna, K. T. (2021). robotics. WhatIs.com. https://www.techtarget.com/whatis/definition/robotics.
2. Ayaida, M.; Messai, N.; Valentin, F.; Marcheras, D. TalkRoBots: A Middleware for Robotic Systems in Industry 4.0. *Future Internet* 2022, *14*, 109. https://doi.org/10.3390/fi14040109Kossinets, Gueorgi, and Duncan J. Watts. 2009. "Origins of Homophily in an Evolving Social Network." American Journal of Sociology 115:405–50. Accessed February 28, 2010. doi:10.1086/599247.
3. Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Rajiv Suman, Substantial capabilities of robotics in enhancing industry 4.0 implementation,Cognitive Robotics,Volume 1, 2021, Pages 58-75, ISSN 2667-2413,https://doi.org/10.1016/j.cogr.2021.06.001. (https://www.sciencedirect.com/science/article/pii/S2667241321000057).
4. St-Onge, D., Herath, D. (2022). The Robot Operating System (ROS1 &2): Programming Paradigms and Deployment. In: Herath, D., St-Onge, D. (eds) Foundations of Robotics. Springer, Singapore. https://doi.org/10.1007/978-981-19-1983-1_5
5. Marín, Pablo & Hussein, Ahmed & Martín Gómez, David & de la Escalera, Arturo. (2018). Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles. Journal of Advanced Transportation. 2018. 1-10. 10.1155/2018/6392697.
6. Peng, M. (2023, June 19). RPlidar A3 - SLAMTEC Global Network Multi-point Interaction Preferred RPlidar A3 LiDAR Lidar Sensor. SLAMTEC Global Network. https://www.slamtec.com/en/Lidar/A3/