Indoor Environment Monitoring in Search of Gas Leakage by Mobile Robot

João Braun[1], Luis Piardi[1,2(✉)], Thadeu Brito[2], José Lima[2,4], Ana Pereira[2], Paulo Costa[3,4], and Alberto Nakano[1]

[1] Federal University of Technology - Paraná (UTFPR), Toledo, Brazil
jneto.1996@alunos.utfpr.edu.br, nakano@utfpr.edu.br

[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI) and IPB, Bragança, Portugal
{fpiardi,brito,jllima,apereira}@ipb.pt

[3] Faculty of Engineering of University of Porto (FEUP), Porto, Portugal
paco@fe.up.pt

[4] Centre for Robotics in Industry and Intelligent Systems - INESC TEC, Porto, Portugal

Abstract. Inspection based on mobile autonomous robots can assume an important role in many industries. Instead of having fixed sensors, the concept of assembling the sensors on a mobile robot that performs the scanning and inspection through a defined path is cheaper, configurable and adaptable. This paper describes a mobile robot, equipped with several gas sensors and a LIDAR device, that scans an established area by following a trajectory based on way-points searching for gas leakage and simultaneously avoid obstacles in the map. In other words, the robot follows the trajectory while the gas concentration is under a defined value and surrounding the obstacles. Otherwise, the autonomous robot starts the leakage search based on a search algorithm that allows to find the leakage position. The proposed methodology is verified in simulation based on a model of the real robot. The search test performed in a simulation environment allows to validate the proposed methodology.

Keywords: Algorithm gas search · Simulation · Autonomous mobile robot · LIDAR

1 Introduction

Nowadays robots can be used in several contexts such as inspection of industrial plants, search for environmental pollutant sources, explosives and drugs at airports and harbors. In the context of gas leakage, robots can be used for inspections in several structures, mainly in industrial plants, to prevent harm to humans and nature, and avoiding high financial losses [1]. The need of finding

M. F. Silva et al. (Eds.): ROBOT 2019, AISC 1093, pp. 339–350, 2020.
https://doi.org/10.1007/978-3-030-36150-1_28

the leak source without exposing humans to hazardous environments and accessing remote locations is what drives many researches in robotized leak detection [2].

In this context, robots can map any area while searching for predetermined levels of gases. In order to do this, the robot must be equipped with sensors with the ability to measure the gas emission, to locate itself, to plan the route to an objective point and to deviate from known and unknown obstacles. Developing a method that accomplishes these tasks can help industrial sectors avoid potential risks to workers. Therefore, the purpose of this work is to implement a method that meets these requirements through a simulation approach, to avoid possible failures in the real system. With the simulated environment, it is possible to test and validate the proposed gas search algorithms to perform the mentioned tasks, simulating adverse situations in the environment for virtual robot decision making. The SimTwo simulator arises a friendly way to simulate a 3D environment, the mobile robot and the gas emission distribution from a source [3].

This work is structured as follows. After the introduction, a brief state of art is presented in Sect. 2. Then, in Sect. 3, the robot model is stated showing the similarities and differences between the real robot and the simulated one. The algorithms and the gas distribution model used in this work are explained in Sect. 4. The project results are shown in Sect. 5. The conclusion and future work are discussed in Sect. 6.

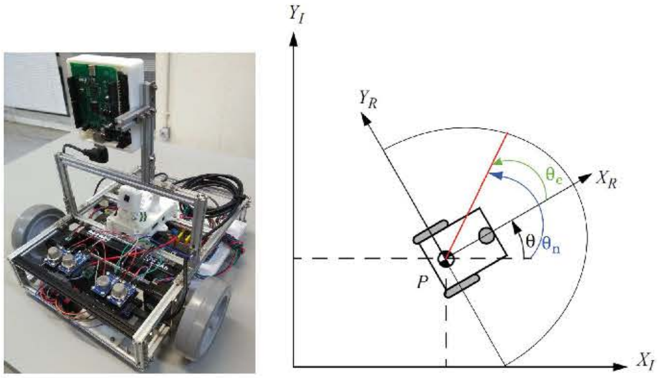
2 State of Art

Studies and development in mobile robotics have steadily increased over the years. Of all the possible tasks a mobile robot can do, having gas sensors is a great way to detect gas leakage, that is, robots with a "sense of smell" [4]. Detecting gas leakage can also be performed by more than one robot, as [5] demonstrates, by applying the Kalman filter it is possible to command five robots to find a certain concentration of gas.

The gas detection response is directly influenced by the discrimination of the gases to be detected, then [6] demonstrates an approach that inserts an array of sensors into the mobile robots. In this sense, each sensor is responsible for identifying a specific type of gas, speeding up the identification process. Another approach of multiple sensors to discriminate gases can be seen in [7], that develop a system with sensors that together distinguish different concentrations of propane, acetone, and ethanol. In order to detect large scale gas, [8] applies the decentralized Gas Distribution Map (GDM) method. Generating a Hilbert map through probabilistic representations, it addresses the task of finding gas concentration in the multiple classes. In this way, it points to an alternative way to GDM to map an environment with measurements learned from the place. As the mobile robot advances into the environment, it is possible to automatically perform unsupervised learning through the system developed by [9]. In an on-the-fly way the system discriminates the gases present in the site, which can serve as an extra tool for rescue teams.

3 Robot Model

In order to validate the approach and algorithms for the inspection and gas leakage detection, the structure of a real mobile robot already developed, will be simulated (seen in Fig. 1a). For further information about the robot, such as its localization system, the reader is referred to [10–13]. The Hokuyo Laser Ranger Finder (LRF) was modeled in SimTwo and validated in [14]. This model, applied in this project, presents noise similar to the real device. In this way, it is possible to obtain results near to the real sensor. This transducer detects obstacles in two dimensions via Time of Flight technology. The real configuration setup was applied in simulation: number of laser beams, the laser position in the free space $[x, y, z]$, its angle in the free space, the area coverage (180° semi circle with 3–400 cm radius), sample period and the noise. In this way, the sensor area of coverage is illustrated in Fig. 1b. After the validation in simulation presented in this work, the robot will be tested in a real controlled scenario.



(a) Real robot.

(b) LIDAR's area of coverage, modified from [15].

Fig. 1. Real structure of the mobile robot.

The software used to create the simulation environment is the SimTwo [3]. This software presents a realistic model of the 3D environment, which represents the dynamic constraints that exist in the real environment. In this sense, with the environment created through the simulator, is possible to validate the approach for the detection and search of gas-emitting sources, and then, in a future stage of the work, to carry out the tests in real environment.

4 Path Planning and Gas Search Algorithm

To search for the gas leakage avoiding obstacles not only the gradient search algorithm is needed but also a path planning. The path planning is used to reach the points that the gas search algorithm computed in an optimal way, without colliding with the obstacles. Binary Heap A* is utilized for the path planning, based on [16]. First, the user elaborates an inspection trajectory for the robot by selecting way-points in the developed control application. After, the application will interpolate a spline function with these points, as it can be seen in the next Section. The SimTwo simulator sends the robot pose and the gas concentration, $[x, y, \theta, z]_{Actual}$, to the application. The variables of communication between these processes are presented in Fig. 2.

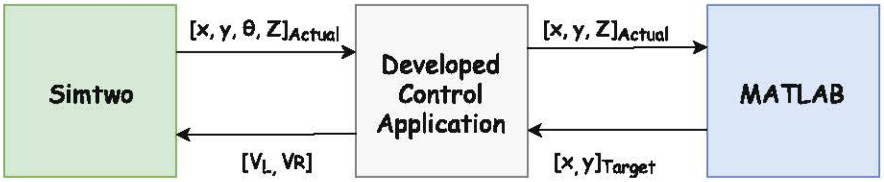


Fig. 2. System communication structure.

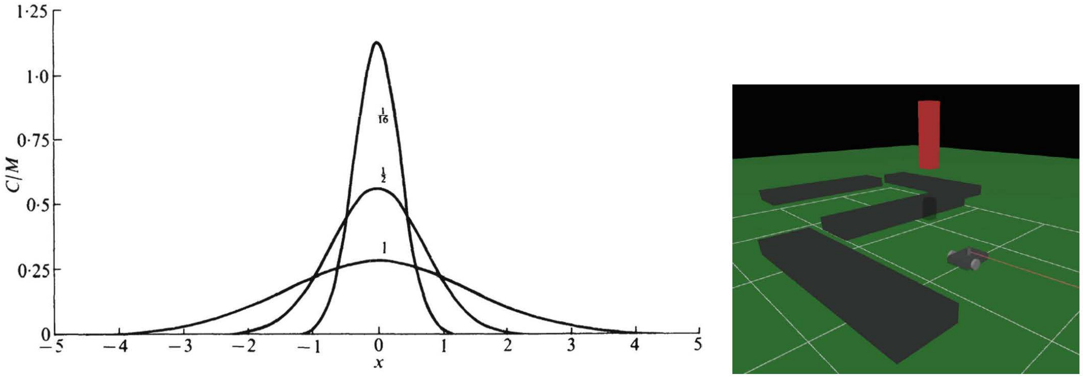
With this information, and sampling the spline function, an algorithm (already presented in [17]), makes the robot follow the trajectory computed by the interpolation. The sampling criterion is performed by the Euclidean distance between points that belong to the curve. During all the process, the robot keeps measuring gas substances. In simulation, the system simulates the leak through a gas model. The gas concentration (Z) model for a constant diffusion can be seen in [18]

$$Z(x,y) = \frac{M}{2 \cdot \sqrt{(\pi \cdot D \cdot t)}} \cdot e^{-\frac{(h)^2}{4 \cdot D \cdot t}} \tag{1}$$

given by Eq. (1) for the position of the robot (x, y). Where M represents the amount of substance deposited at time $t = 0$ at $[x0, y0]$; D is the diffusion constant; t the time and h is $\sqrt{(x - x0)^2 + (y - y0)^2}$, i.e, the Euclidean distance between the measure position $[x, y]$, and $[x0, y0]$ the gas peak position.

As one can see, this model is time dependent, at each time the model represents another concentration value. However, the robot converges rapidly to the gas model peak position. Moreover, the diffusion is slow configured by parameter D. Thus the time influence was discarded by choosing a constant value for t. Figure 3a denotes the concentration values in one dimension, for several values of $D \cdot t$.

It is assumed an indoor environment so the influence of the wind is discarded. In future work, it will be addressed the wind disturbance as well as noise in gas acquisition. The simulated environment can be seen in Fig. 3b where the red cylinder is used to represent the gas leak. Hereupon, the SimTwo will simulate the gas sensors measurements by calculating the gas concentration at the actual robot position. Note that, the gas leak source is simulated assuming a depressurized leakage, in this way, nothing will influence the shape of the distribution. If the concentration pass a certain threshold (defined by the user with range from 0 to 100), the system will discard the user defined trajectory, and will start to seek for the gas source.



(a)

(b)

Fig. 3. Figure (a) represents the gas model in one dimension for several $D \times t$ and x values. Figure (b) displays the simulation environment.

The proposed algorithm for gas search is running on MATLAB and the communication variables can be seen in Fig. 2 that uses UDP Ethernet datagrams. The application will send the robot pose and the concentration $[x, y, \theta, z]_{Actual}$ to MATLAB that process the search algorithm for the leak localization. By this way, MATLAB sends the next position $[x, y]_{Target}$ with high probability of a higher concentration to probe. MATLAB was chosen to run the search tool, as it is easier to implement more complex algorithms.

The MATLAB implemented search algorithm is described as follows. Let $[x, y]_{Actual}$ be the robot actual position and $[x, y]_{Previous}$ the robot previous position. Consider d the previous direction took by the robot, defined by $d = [x, y]_{Actual} - [x, y]_{Previous}$ and consider $\bar{d}$ the direction between the robot actual position and the position where was identified the higher gas concentration ($[x, y]_{Higher}$) and can be defined as $\bar{d} = [x, y]_{Higher} - [x, y]_{Actual}$. Define $Z_{Previous}$, Z_{Actual} and Z_{Higher} the gases concentrations at $[x, y]_{Previous}$, $[x, y]_{Actual}$ and $[x, y]_{Higher}$, respectively.

If $Z_{Previous} \leq Z_{Actual}$, the robot target position will be defined as

$$[x, y]_{Target} = \begin{cases} [x, y]_{Actual} + d, & \text{if } Z_{Higher} \leq Z_{Actual}; \\ [x, y]_{Actual} + d + \bar{d}, & \text{otherwise.} \end{cases} \quad (2)$$

If $Z_{Previous} > Z_{Actual}$, the robot target position will be in the opposite direction, that means, if d is a horizontal movement then the new direction will be the opposite diagonal direction. In the same way, if d is a diagonal direction then the new direction will be the opposite horizontal direction. When the procedure take an opposite direction the choice to take right/left depends of the condition of Z_{Higher}, that is, if Z_{Higher} is greater (or lower) than the Z_{Actual}. In this way, soon after the robot quits the inspection trajectory, the robot will search for a direction with a greater gas concentration and will keep moving in the same direction until the gas concentration decreases. When this happens, it will search for the next direction where the gas concentration is higher, and so on.

The described algorithm takes the robot to a near location of the source. This algorithm makes no assumption of the exact gas model 1 and will generate a set of target points and gas concentration measures with increasing concentration values.

5 Results

Several tests were evaluated to validate the inspection route, the path planning, the obstacle avoidance system and the gas search algorithm (with and without the obstacles). In this sense, they will be presented in this Section in the following order: Inspection route, Obstacle avoidance, Gas search algorithm without obstacles and Gas search combined with the first and second items.

5.1 Inspection Route

The user can select control way-points so that the control application can interpolate spline functions that pass through all the control points. Soon after, the control application samples the route, as the robot has a point following controller. The sampling happens every 20 cm. Thus, the path followed by the robot has a smooth performance and by this way a faithful representation to the original spline. The sampling has a low pass characteristic as can be seen in Fig. 4, especially by the white arrows indicators.

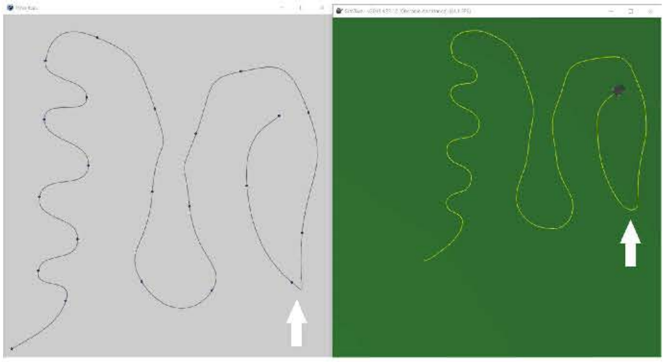
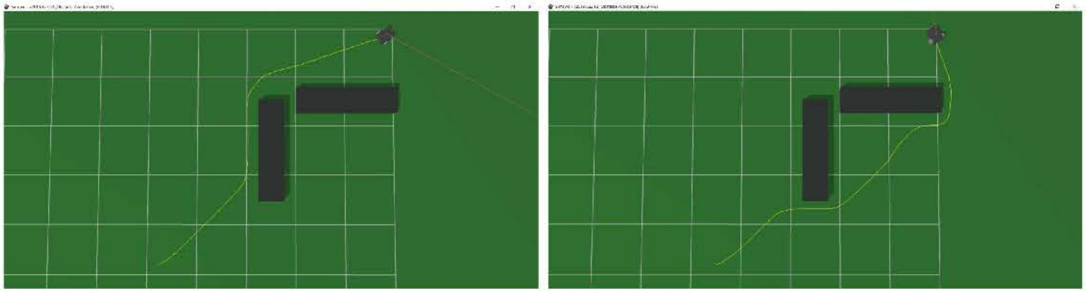


Fig. 4. Application screenshot of the planning trajectory inserted by way-points from the user.

The robot's controller is developed in such a way that whenever the robot gets near the actual point, the control application sends another point. It is repeated until the last point, where the robot follows a deceleration function to stop to the last point. As the controller is not on this scope of this work, the developed controller details can be seen in [11].

5.2 Obstacle Avoidance

Two scenarios were tested to prove the path planning algorithm developed for this work. In the first scenario, the robot is inserted near the origin $[x, y] = [0.2, 0.2]$ and set to go to the point $[x, y] = [5, 5]$. Then, is added two obstacles blocking the direct path to the target. Note that, the path planning algorithm developed is based in Binary Heap A* method, that uses a proportionality constant to generate sub optimal paths in less time [19]. Thus, it is difficult to predict if the robot will choose the shortest path. Figure 5a displays the first scenario.



(a) First position.

(b) Second position.

Fig. 5. Path planning test scenario using a Binary Heap A*. The grid does not represent the map limitations.

In another test, the robot is inserted a little farther to the right $[x, y] = [0.5, 0.2]$, to force the robot to choose the right path, as the heuristic cost to reach the target point is lower. Bear in mind that the LIDAR model works from 3 cm to 400 cm. In this concept, the robot can not see obstacles beyond this range. Thus, the robot continues the right path until it sees the rightmost obstacle. Although, when it sees this obstacle, the heuristic cost to go back to the left path is still higher. Thus, the robot continues with the first path choice. Figure 5b presents this behaviour. Consider that the grid is just illustrative, it does not mean the maps limitations.

Finally, in the second scenario, two more obstacles are placed to jam the path planning. As can be seen in Fig. 6, the robot is placed in $[x, y] = [1.5, 0.2]$ to decrease even further the heuristic cost to the robot perform the right path. However, as the robot can not see beyond 400 cm, the robot does not know the obstacles until they reach its range. In this test, the robot keeps going to the farthest path even though the left one was the shortest. Making the robot circle all the obstacles.

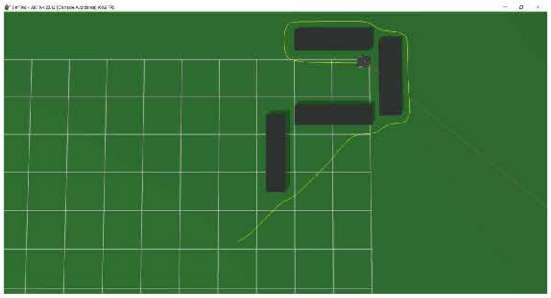
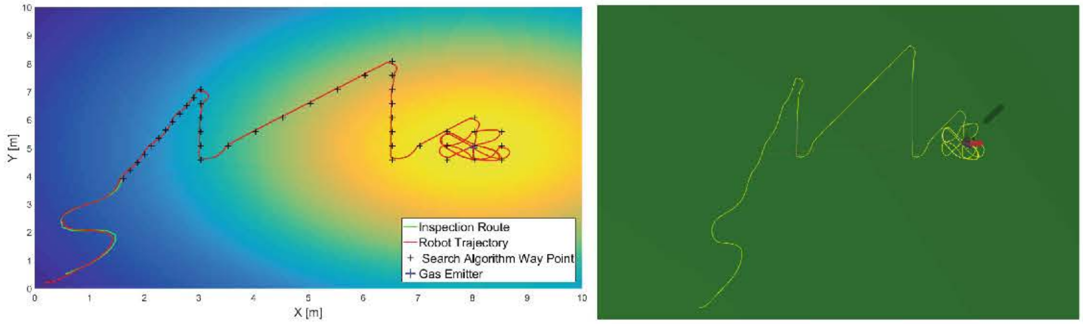


Fig. 6. Path planning test second scenario using a Binary Heap A*. The grid does not represent the map limitations.

5.3 Gas Search Algorithm Without Obstacles

As explained in Sect. 4, the algorithm does not takes assumption on the gas model and its maximum concentration value, as in the real scenario this can not be predicted. In this way, the algorithm is tested without obstacles in just one scenario, because the main objective of this work is to validate the algorithm with the obstacle avoidance feature. The gas position model in this test can be seen in Fig. 10.

Moreover, Fig. 7a illustrates the comparison between the path performed by the robot and the route made by the user in the application as well as the gas search algorithm points.



(a) Comparison of the inspection route and gas search algorithm.

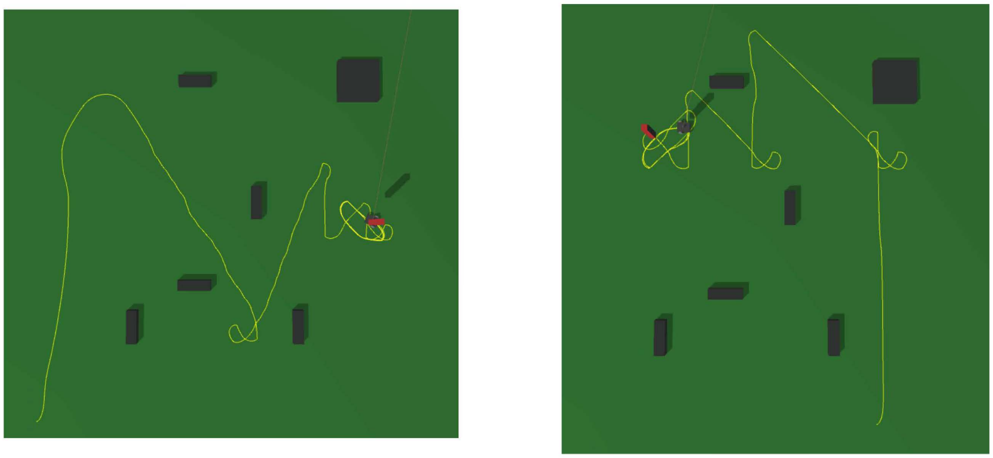
(b) Path planning created in SimTwo.

Fig. 7. Path planning test along with gas search algorithm without obstacles.

Figure 7b presents the simulation result that was represented by Fig. 7a. Note that, when the robot abandons the inspection route and starts to search for the gas leak, the black plus symbols in Fig. 7a starts to appear in the plot (which are the points computed by the gas search algorithm), i.e, the robot follows the points sent by the gas search algorithm. Moreover, it is important to note the similarity between the path performed by the robot and the path generated by the simulation. Little differences can be seen because of the spline sampling and the controller. In addition, as explained, the search algorithm behaviour makes the robot keep going until the gradient becomes negative, then it changes its direction. Finally, when it reaches the maximum point, it keeps circling the peak value. It was decided to not implement a stop point as the idea behind this work is to find the gas leak. Therefore, the robot keeps circling it.

5.4 Gas Search with Obstacle Avoidance

Finally, two challenging scenarios were tested with the gas algorithm combined with the path planning. The first analysis is applied to obstacles in the same gas model position in the previous verification. Figure 8a displays the robot and the algorithms behaviour. In this concept, the inspection route is drawn (green line in Fig. 9) to the robot start the test.



(a) Gas search algorithm with obstacles first scenario test.

(b) Gas search algorithm with obstacles in the second scenario test.

Fig. 8. Gas search tests with obstacles.

As can be seen in Figs. 8a and 9, the robot could perform the search with several obstacles placed in the scenario. Moreover, note that the robot followed the gas search points smoothly (black plus symbols in Fig. 9). Additionally, whenever the gas search algorithm sent a point that the robot could not reach, the control application with the path planning, redirected the robot to the nearest point accessible. The robot with the algorithms converged to the peak without any problems. Figure 10 displays the gas model in the first test.

In the second scenario the gas model position is changed to $[x, y] = [2.5, 7]$ and the robot to $[x, y] = [7.5, 0.5]$. Figures 8b and 11 displays the robot's path behaviour in the second scenario with the obstacles.

As can be seen, during the second test, the robot could perform the search without colliding with any obstacles. The gas search algorithm behaviour is well

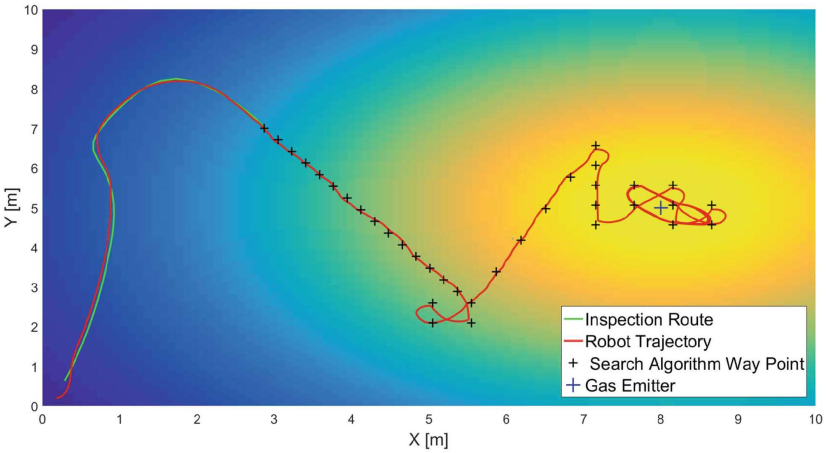


Fig. 9. Inspection route, path performed by the robot and the algorithm search points in a flat plot during the first scenario.

seen in the change of directions in the robot's path, represented by the yellow line in Fig. 8b. As explained in Sect. 4, the robot keeps moving in the direction that has a higher positive gradient and, when it becomes negative, it changes direction. Therefore, soon after the robot quits the inspection route (when the black plus symbols starts to appear), the robot keeps going north until the gradient becomes negative and then changes its direction, and this happens two more times, until the robot finds the gas leak and keeps circling it.

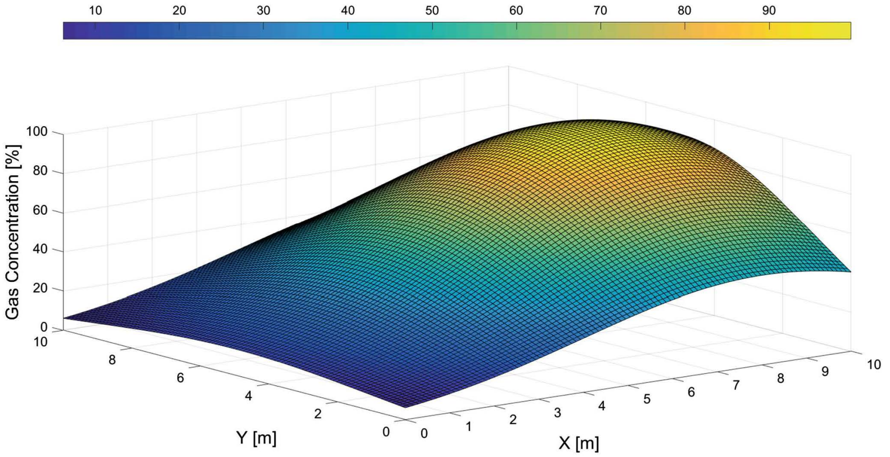


Fig. 10. Gas model in the first scenario in the combined test.

Again, the green line, which is the inspection route and the red line, which is the path performed by the robot, are similar (seen in Fig. 11). This displays that the robot's controller is well adjusted.

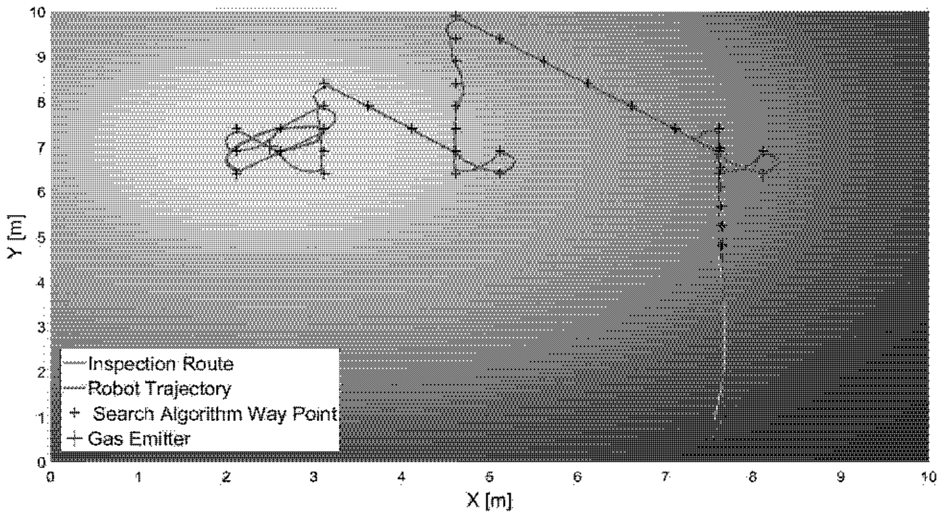


Fig. 11. Inspection route, path performed by the robot and the algorithm search points in a flat plot during the second scenario.

6 Conclusion and Future Works

This work presented and validated through realistic simulation an autonomous mobile robot that could perform an inspection route selected by the user measuring for gas substances. Moreover, if a certain gas threshold value passed, the robot would abandon the trajectory and search autonomously for the gas source, avoiding obstacles simultaneously. In addition, the gas model, the obstacle avoidance system, path planning and data gathering techniques are presented. The gas search algorithm proved to be well suited for the indoor simulation environment, and combined with the obstacle avoidance algorithm, the tests showed promising results. Finally, the path similarity between the inspection route and the path performed by the robot are very similar, proving the controller is well adjusted in simulation. In future work, the idea is to test this in a real environment.

Acknowledgements. This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project (POCI-01-0145-FEDER-006961), and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

1. Baetz, W., Kroll, A., Bonow, G.: Mobile robots with active IR-optical sensing for remote gas detection and source localization. In: 2009 IEEE International Conference on Robotics and Automation, pp. 2773–2778. IEEE (2009)
2. Kowadlo, G., Russell, R.A.: Robot odor localization: a taxonomy and survey. Int. J. Robot. Res. **27**(8), 869–894 (2008)

3. Costa, P., Gonçalves, J., Lima, J., Malheiros, P.: SimTwo realistic simulator: a tool for the development and validation of robot software. Theory Appl. Math. Comput. Sci. **1**(1), 17–33 (2011)
4. Wandel, M., Lilienthal, A.J., Duckett, T., Weimar, U., Zell, A.: Gas distribution in unventilated indoor environments inspected by a mobile robot. In: IEEE International Conference on Advanced Robotics, ICAR 2003, Coimbra, Portugal, 30 June - 3 July 2003, vol. 1, pp. 507–512. University of Coimbra (2003)
5. Marques, L., Martins, A., de Almeida, A.T.: Environmental monitoring with mobile robots. In: 2005 IEEE/RSJ International Conferene on Intelligent Robotics and Systems, pp. 3624–3629. IEEE (2005)
6. Trincavelli, M.: Gas discrimination for mobile robots. KI-Künstliche Intell. **25**(4), 351–354 (2011)
7. Xing, Y., Vincent, T.A., Cole, M., Gardner, J.W., Fan, H., Bennetts, V.H., Schaffernicht, E., Lilienthal, A.J.: Mobile robot multi-sensor unit for unsupervised gas discrimination in uncontrolled environments. In: 2017 IEEE SENSORS (2017)
8. Zhu, P., Ferrari, S., Morelli, J., Linares, R., Doerr, B.: Scalable Gas sensing, mapping, and path planning via decentralized hilbert maps. Sensors **19**(7), 1524 (2019)
9. Fan, H., Hernandez Bennetts, V., Schaffernicht, E., Lilienthal, A.J.: Towards gas discrimination and mapping in emergency response scenarios using a mobile robot with an electronic nose. Sensors **19**(3), 685 (2019)
10. Lima, J., Costa, P.: Ultra-wideband time of flight based localization system and odometry fusion for a scanning 3 DoF magnetic field autonomous robot. In: Iberian Robotic Conference, pp. 879–890. Springer, Cham (2017)
11. Piardi, L.: Application of a mobile robot to spatial mapping of radioactive substances in indoor environment. Master's thesis. IPB (2018)
12. Piardi, L., Lima, J., Costa, P.: Development of a ground truth localization system for wheeled mobile robots in indoor environments based on laser range-finder for low-cost systems. In: ICINCO, no. 2, pp. 351–358 (2018)
13. Braun, J., Brito, T., Lima, J., Costa, P., Costa, P., Nakano, A.: A Comparison of A* and RRT* algorithms with dynamic and real time constraint scenarios for mobile robots. In: Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, vol. 1, pp. 398–405. SIMULTECH (2019). ISBN 978-989-758-381-0. https://doi.org/10.5220/0008118803980405
14. Lima, J., Gonçalves, J., Costa, P.J.: Modeling of a low cost laser scanner sensor. In: CONTROLO'2014–Proceedings of the 11th Portuguese Conference on Automatic Control, pp. 697–705. Springer, Cham (2015)
15. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots. MIT Press, Cambridge (2011)
16. Sheikh, T.S., Afanasyev, I.M.: Stereo vision-based optimal path planning with stochastic maps for mobile robot navigation. In: International Conference on Intelligent Autonomous Systems, pp. 40–55. Springer, Cham (2018)
17. Piardi, L., Lima, J., Costa, P., Brito, T.: Development of a dynamic path for a toxic substances mapping mobile robot in industry environment. In: Iberian Robotic Conference, pp. 655–667. Springer, Cham (2017)
18. Crank, J.: The Mathematics of Diffusion. Oxford University Press, Oxford (1979)
19. Lima, J., Costa, P., Costa, P., Eckert, L., Piardi, L., Moreira, A., Nakano, A.: A* Search Algorithm Optimization Path Planning in Mobile Robots Scenarios. In: International Conference of Numerical Analysis and Applied Mthematics (2018)