



Autonomous Robot Navigation for Automotive Assembly Task: An Industry Use-Case

Héber Sobreira¹, Luís Rocha¹, José Lima^{2(✉)}, Francisco Rodrigues¹,
A. Paulo Moreira³, and Germano Veiga³

- ¹ INESC TEC - INESC Technology and Science, Porto, Portugal
{heber.m.sobreira,luis.f.rocha,francisco.a.rodrigues}@inesctec.pt
- ² INESC TEC - INESC Technology and Science and CeDRI - Research Centre
in Digitalization and Intelligent Robotics, Polytechnic Institute of Bragança,
Bragança, Portugal
jllima@ipb.pt
- ³ INESC TEC - INESC Technology and Science and Faculty of Engineering
of University of Porto, Porto, Portugal
amoreira@fe.up.pt, germano.veiga@inesctec.pt

Abstract. Automobile industry faces one of the most flexible productivity caused by the number of customized models variants due to the buyers needs. This fact requires the production system to introduce flexible, adaptable and cooperative with humans solutions. In the present work, a panel that should be mounted inside a van is addressed. For that purpose, a mobile manipulator is suggested that could share the same space with workers helping each other. This paper presents the navigation system for the robot that enters the van from the rear door after a ramp, operates and exits. The localization system is based on 3DOF methodologies that allow the robot to operate autonomously. Real tests scenarios prove the precision and repeatability of the navigation system outside, inside and during the ramp access of the van.

Keywords: Autonomous robot · Navigation · Human cooperation

1 Introduction

Automobile industry presents one of the most flexible productivity caused by the number of models variants due to the customer needs. Moreover, the workers' ergonomics should be attended. This demanding pushes the manufacturers to look for new solutions that increase the flexibility, reduce the production costs and also finds a better ergonomic posture to the worker. A particular inconvenient for the worker is the assembly operations inside the vehicle, where the human worker has to go inside/outside the vehicle several times per shift and many assembly operations are near the floor of the vehicle. The posture inside

the vehicle (crouch) promotes injuries. These issues encourage researchers to propose solutions that solve the problems in a collaborative way with workers. This paper presents a solution, based on a mobile manipulator, that navigates a mobile robot, in an autonomous way, to enter the van from the rear door. It also addresses the localization system and validates it in a real scenario. It demonstrates the integration of advanced methodologies of localization, navigation and control for mobile robots. The development of this heterogeneous ecosystem of navigation and localization solutions. This paper is organized as follows: after a brief introduction, the state of the art of the industrial localization systems is presented. Then, Sect. 2 addresses the use-case scenario and describes the motivation for this work. In Sect. 3, the adopted mobile robot platform is presented as well as its hardware and software components. Sections 4 and 5 address the localization and navigation methodologies for the developed system. Section 6 evaluates the results through the precision and the repeatability of the navigation system. Finally, Sect. 7 concludes the paper and points out some future work direction.

1.1 State of the Art

Actually, Industrial mobile robots (AGVs, Automatic Guided Vehicle), can self-localize and move autonomously without human intervention. They are used to transport materials between work stations in warehouses and production lines.

AGVs are used in industrial environments for more than 50 years and both the algorithms and hardware used has been evolving in order to increase the accuracy, robustness and flexibility while decreasing costs of the overall system. Although, we are focusing in industrial autonomous robots (AGVs), the localization problem is transversal to all indoor autonomous robots application areas. Regarding the localization systems applied to the industrial mobile robots, it is common to use several solutions such as [1,2]:

- Wire Guidance (following a buried cable in the floor)
- Strip Guidance (magnetic or colored strips arranged on the floor, line detection is performed by hall effect sensors or optical sensors)
- Based on Marker (embedded in the ground, markers can be magnetic labels, reflectors, passive RF, geometric shapes or bar codes)
- Trilateration and Triangulation (detecting the localization of the robot through beacons usually arranged in high parts of the walls with a laser)

Meanwhile, in the last decade localization based on natural marks has been increasing [3,6]. These natural marks are composed by a set of distances and angles to the detected objects (such as doors, walls, furniture, etc.) that can be acquired through an on-board laser range finder. This method has the main advantage of not requiring the installation of dedicated reflectors in the environment, which in some factories might not be a viable option. On the other hand, it is expected that even without special markers and straight corridors, the localization system remains robust. Besides these advantages, this approach

needs to process a significant amount of sensor data efficiently in order to provide real-time localization. Therefore, the map-matching algorithms must be optimized in terms of accuracy, processing time, convergence speed and also sensor noise robustness. The map-matching is a method of self-localization for mobile robots in which the local environment map (actual data acquired by the robot) is matched with an already stored map. Authors have worked with several industry applications based on the perfect match algorithm [7, 8, 10]. With these topics in mind, the paper addresses the localization and navigation of a mobile platform (able to perform assembly tasks) that allow the robot to move inside the van through a ramp and position itself to operate autonomously.

2 Use-Case Description

A high dynamic environment characterizes this use case scenario where the presence of Human operators is constant. Here, the main requirements for the navigation system are to be reliable, dynamic and adaptable to the real environment. The proposed solution, based on a mobile manipulator, is composed by the localization system that allows the robot, in an autonomous way, to navigate and enter the van from the rear door. Previously, the mobile manipulator carries a kit of screws that will be delivered to the worker which is inside the van. Once inside, the wheeled robot should create the path and positioning itself where its manipulator reachability accesses the screw position. The localization can be done resorting to different approaches such as 3DoF and 6 DoF. The 3DoF localization were done with a real robot in real scenario (although a simplified one with planar surface). Navigation problem can be addressed through four different approaches:

- Navigation in the factory (outside the van)
- Navigation in the ramp
- Navigation in the ramp-van transfer
- Navigation inside the van

3 Mobile Platform Description

In this section we make a short introduction to the mobile platform used, detailing the hardware and software configuration used to test the developed navigation and localization algorithms.

3.1 Hardware Description

For our test on a real scenario we used a commercial mobile platform, which was built on top of a Husky UGV, an outdoor research robot from Clearpath Robotics [4]. It has a size of $990 \times 670 \times 390$ mm and a maximum speed of 1,0 m/s.

For the purpose of our localization and navigation tests we assembled a Sick laser LSM151 in Husky's robot front. This laser has an aperture angle of 270° and an operating range of 50 m with a scanning frequency of 25/50 Hz and an angular resolution of $0.25/0.5^\circ$ (Figs. 1 and 2).



Fig. 1. ColRobot platform, based on a husky platform with UR10 arm attached [4,5].

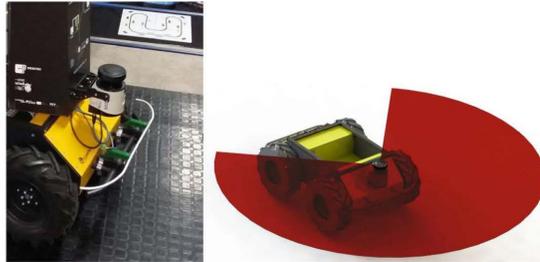


Fig. 2. Attached laser Sensor (LSM151 Sick) used in the experiment.

3.2 Software Description

Concerning the software architecture, we can divide it into several modules: (i) the localization, (ii) the decision, (iii) the controller, and (iv) the ground truth (Fig. 3).

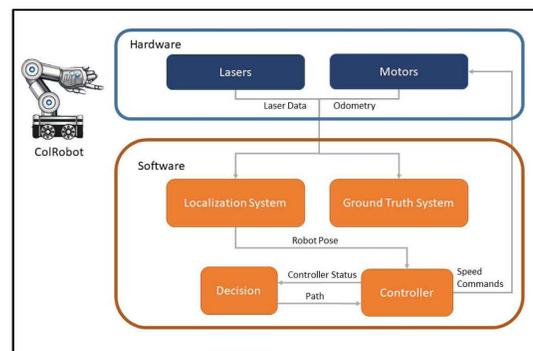


Fig. 3. The orange rectangle represents the software whereas the blue rectangle the hardware. Different modules and their interaction.

The localization system, (i), has the responsibility to determine the pose of the robot in the environment. It uses as input both the data from the Sick laser range finder and the odometry from the vehicle wheels. In this module we use a map matching algorithm, the Augmented Perfect Match, along with a map switch approach. When the robot is outside the van, we use a 2D map of the environment, built using SLAM, to determine its position. When inside the van, we use only the inner counters of the van interior, also pre acquired

using SLAM, for the self-localization of the robot. Restricting the map to the van, allows the system to increase the positioning accuracy, which is important for the subsequent arm screwing operation. The map transition determined by the position of the robot path.

The decision module, (ii), is responsible for the definition of the robot trajectory, which is computed based on a fixed graph built on top of the environment map.

The controller, (iii), is responsible for guiding the robot in order to perform a trajectory. It uses as input the pose of the robot determined by the localization system.

The ground truth system (iv), has as purpose to determine the real pose of the robot with high precision. With it, we can estimate the error related to the navigation system, composed by the localization system and the controller.

4 Localization System

To solve the robot localization problem, for the use-case presented earlier, we analyzed several algorithms namely the: Augmented Perfect Match (APM), the Iterative Closest Point (ICP) and the Normal Distribution Transform (NDT). These algorithms were compared using different metrics and we have concluded that the APM is lighter in terms of computational weight, and also presents higher tolerance to orientation errors, making it a very interesting approach for the problem at hands. For more detail about this comparison please refer to [11].

Based on this conclusions we decided to use the APM for the navigation outside the van, since in this scenario the precision requirements are not so hard. Inside the van, and only if higher precision is needed, we propose to use a 6 DoF localization algorithm, LUT-ICP [11], allowing in this way to achieve a lower positional error of the robotic arm, important for some operations.

In more detail, the 3DoF localization system uses the result of APM, as a laser observation measurement, and fuses it with the vehicle's odometry data using an Extended Kalman Filter (EKF). The algorithm of Matching is based on the light computational Perfect Match algorithm, described by Lauer et al. in [12]. In this algorithm the vehicle pose is computed using 2D distance points from the surrounding environment. These points are acquired with a laser range finder and are matched with the map of the building previously computed. Therefore, the vehicle pose is calculated by trying to minimize the fitting error between the data acquired and the environment map. For details, see [9].

Despite of the solution presented, during the tests in real scenario, and in what concerns the localization, we verified that the 3 DoF localization system achieved sufficiently good results (as presented in Results section), guaranteeing the minimum requirements so that the robotic arm can carry out the remaining operations.

5 Parametric Trajectory Controller

The trajectory controller block determines the speed of wheels that allow to follow the desired trajectory, in a closed loop way, as presented in Fig. 4, based on the robot pose. During the phase where the robot is climbing to the inside of the van, it will perform a fixed trajectory. Such will avoid potential hazards for both the robot and human operators in the area. Therefore, we will use a path-following controller.



Fig. 4. Controller inputs and outputs.

The *PoseRob* represents the pose and orientation of the robot related to the absolute referential of the navigation:

$$PoseRob = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$

The *Trajectory* is composed by two parametric equations (F_x, F_y) which define a set of reference positions related to the absolute referential of the navigation (see Fig. 5). F_x and F_y are two n -order polynomials, which define such positions through the parameter t . The starting point of the trajectory corresponds to $t=0$ and the end point corresponds to $t=1$.

$$Trajectory(t) = \begin{bmatrix} F_x(t) \\ F_y(t) \end{bmatrix} : t \in [0 \ 1] \quad (2)$$

$$F_x = \sum_{i=0}^{n-1} A_i * t^i \quad (3)$$

$$F_y = \sum_{i=0}^{n-1} B_i * t^i \quad (4)$$

The *ControlRob* is the control command sent to the hardware and is composed by linear (V) and angular velocity (W).

$$ControlRob = \begin{bmatrix} V \\ W \end{bmatrix} \quad (5)$$

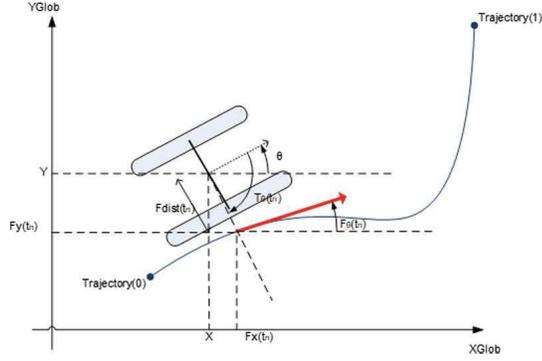


Fig. 5. Controller trajectory.

A path-following controller tends to minimize two types of errors. The first one that represents the distance between the robot and the path, and the second one related to the difference between the orientation of the robot and the orientation of the path.

The variable t_n represents the parameter t which minimizes the distance between a position on the trajectory and the pose of the robot.

$F_{Dist}(t_n)$ is the distance error between the robot and the trajectory:

$$F_{Dist}(t_n) = \sqrt{(F_x(t_n) - x)^2 + (F_y(t_n) - y)^2} \quad (6)$$

$F_\theta(t_n)$ is the reference orientation for the robot defined by $Trajectory(t_n)$

$$F_\theta(t) = \text{Atan2} \left(\frac{\partial F_x(t)}{\partial t}, \frac{\partial F_y(t)}{\partial t} \right) \quad (7)$$

where

$$\frac{\partial F_x(t)}{\partial t} = \sum_{i=1}^n i * A_i * t^{i-1} \quad (8)$$

$$\frac{\partial F_y(t)}{\partial t} = \sum_{i=1}^n i * B_i * t^{i-1} \quad (9)$$

$Err_\theta(t_n)$ is the orientation error between the robot and the trajectory and can be calculated as follows:

$$Err_\theta(t_n) = \text{NormAng}(F_\theta(t_n) - \theta) \quad (10)$$

$T_\theta(t_n)$ is the angle defined by the pose of the robot and the closest point on the trajectory.

$$T_\theta(t_n) = \text{NormAng}(\text{Atan2}(F_x(t_n) - x, F_y(t_n) - y) - \theta) \quad (11)$$

The linear velocity (V) is constant and one of the arguments passed with the trajectory.

The angular velocity (W) is defined as a function of the distance and orientation error and the value of the feedforward ($feedForward(t_n)$).

if $T_\theta(t_n) > 0$:

$$\begin{aligned} w &= K_{P,D} * Err_D(t_n) + K_{I,D} * \int Err_D(t_n) \\ &+ K_{P,\theta} * Err_\theta(t_n) + K_{I,\theta} * \int Err_\theta(t_n) \\ &+ FeedForward(t_n) \end{aligned} \tag{12}$$

else:

$$\begin{aligned} w &= -K_{P,D} * Err_D(t_n) - K_{I,D} * \int Err_D(t_n) \\ &+ K_{P,\theta} * Err_\theta(t_n) + K_{I,\theta} * \int Err_\theta(t_n) \\ &+ FeedForward(t_n) \end{aligned} \tag{13}$$

$K_{P,D}$ and $K_{I,D}$ are parameters regarding the proportional and integral of the controller related to the distance error, while $K_{P,\theta}$ and $K_{I,\theta}$ are related to the orientation error.

Regarding the $feedForward(t_n)$, this is determined by applying the derivative to the following equations:

$$R = \frac{V}{W} \tag{14}$$

where, R is the radius of the trajectory determined by the linear and angular velocity. The radius of the trajectory $Trajectory(t)$ is defined by the following equations:

$$T_{Radius}(t) = \frac{\sqrt{\left(\frac{\partial F_x(t)}{\partial t}\right)^2 + \left(\frac{\partial F_y(t)}{\partial t}\right)^2}}{\frac{\partial F_\theta(t)}{\partial t}} \tag{15}$$

where:

$$\frac{\partial F_\theta(t)}{\partial t} = \frac{\frac{\partial^2 F_y(t)}{\partial t^2} * \frac{\partial F_x(t)}{\partial t} - \frac{\partial^2 F_x(t)}{\partial t^2} * \frac{\partial F_y(t)}{\partial t}}{\left(\frac{\partial F_x(t)}{\partial t}\right)^2 + \left(\frac{\partial F_y(t)}{\partial t}\right)^2} \tag{16}$$

Combining Eqs. 14, 15 and 16 and assuming R is $T_{Radius}(t)$ we have the following definition of $feedForward(t_n)$:

$$\begin{aligned} feedForward(t) &= \frac{V * \left(\frac{\partial^2 F_y(t)}{\partial t^2} * \frac{\partial F_x(t)}{\partial t} - \frac{\partial^2 F_x(t)}{\partial t^2} * \frac{\partial F_y(t)}{\partial t}\right)}{\sqrt{\left(\frac{\partial F_x(t)}{\partial t}\right)^2 + \left(\frac{\partial F_y(t)}{\partial t}\right)^2} \left(\left(\frac{\partial F_x(t)}{\partial t}\right)^2 + \left(\frac{\partial F_y(t)}{\partial t}\right)^2\right)} \end{aligned} \tag{17}$$

6 Results

As a way to evaluate the precision and repeatability of the navigation system inside a van, we conducted a set of experiments, where, the robot executed two trajectories autonomously, allowing it to climb into the van and out of it. When it was inside the van, we activated the ground truth system to determine and evaluate its final pose.

6.1 Ground Truth System Description and Characterization

The ground truth system relied on a beacon-based localization algorithm installed in the van, which resorts to a method implemented by Sobreira et al. [10]. Such system estimates a pose of a robot through an Extended Kalman Filter while using cylindrical beacons and it has a position error of 0.005 m and an orientation error of 0.2° . On our application, we installed four beacons inside the van.

As the presence of elements with high reflectivity may affect the detection of dark objects, we covered the beacons while the robot was climbing into the van. When the robot was inside of it, we uncovered the beacons and determined its final pose.

6.2 Precision and Repeatability Results

The main goal of tests presented in this section was to evaluate the precision and repeatability of the navigation system inside a van. The results from 20 experiments are presented in the following table and figures.

The Table 1 represents the position and orientation maximum errors, the standard deviations and the average of the absolute errors.

Table 1. Position and orientation maximum errors

	X(m)	Y(m)	θ (deg)
Maximum error	0,019	0,014	2,216
Standard deviation	0,005	0,005	0,598
Average of absolute errors	0,004	0,004	0,351

The Fig. 6 represents a histogram of the position error of the X axis, while the Fig. 7 demonstrates the normal distribution of the errors we observed on our experiments.

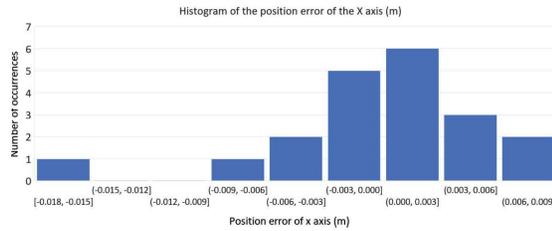


Fig. 6. Histogram of the position error of the X axis.

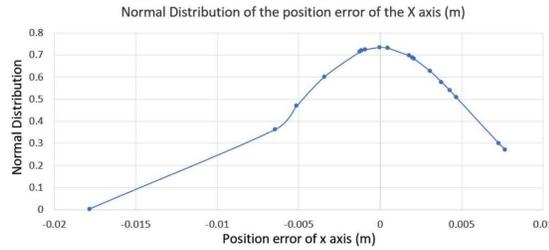


Fig. 7. Normal distribution of the position error of the X axis.

The Fig. 8 represents a histogram of the position error of the Y axis, while the Fig. 9 demonstrates the normal distribution of the errors we observed on our experiments.

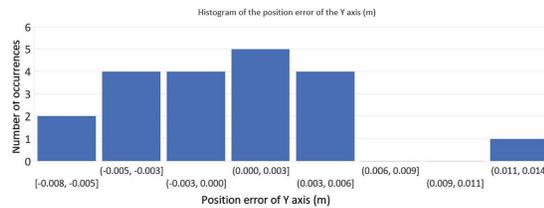


Fig. 8. Histogram of the position error of the Y axis.

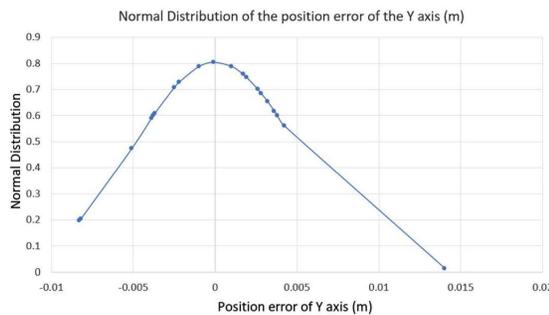


Fig. 9. Normal distribution of the position error of the Y axis.

The Fig. 10 represents a histogram of the orientation errors, while the Fig. 11 demonstrates the normal distribution of the errors we observed on our experiments.

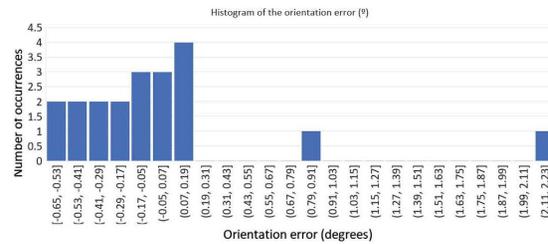


Fig. 10. Histogram of the orientation error.

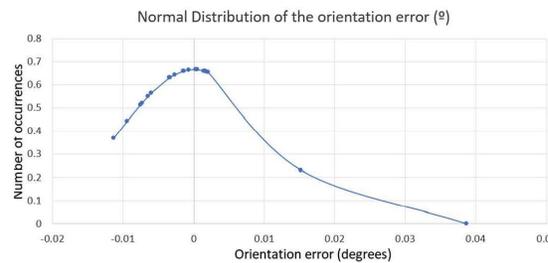


Fig. 11. Normal distribution of the orientation error.

Analyzing Table 1, it is possible to verify that our system had 0.019 m as maximum error in X, 0.014 m in Y and 2.216° in the orientation of the robot. However, observing each figure, we can notice a presence of an outlier, which influenced the maximum error observed for each variable. This outlier was due to the irregularity of the van’s floor. This could have caused the robot to be tilted to one side during the trajectory, which induced in error the localization algorithm. Even though the maximum error was influenced by the presence of each outlier, such errors can be corrected by a vision system with a camera close to the robotic arm end-effector.

7 Conclusion and Future Work

The present work addresses the navigation system for the robot that enters the van from the rear door after a ramp, operates and exits. It is intended that should mounted a panel inside the van. It is suggested a mobile manipulator that could collaborate with workers helping each other. The presented localization system is based on 3DOF methodologies that allow the robot to localize, navigate and operate autonomously. Real tests scenarios prove the precision and repeatability of the navigation system outside, during the access ramp of the van and presents a maximum error of 0.019 m in X, 0.014 m in Y and 2.216° in the orientation.

Acknowledgment. This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project SAICTPAC/0034/2015- POCI-01- 0145-FEDER-016418.

References

1. Schulze, L., Wullner, A.: The approach of automated guided vehicle systems. In: 2006 IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 522–527 (2006)
2. Schulze, L., Behling, S., Buhrs, S.: Automated guided vehicle systems: a driver for increased business performance. In: Proceedings of the International MultiConference of Engineers and Computer Scientists, pp. 19–21 (2008)
3. Pinto, M., Sobreira, H., Moreira, A., Mendonça, H., Matos, A.: Self-localisation of indoor mobile robots using multi-hypotheses and a matching algorithm. *Mechatronics* **23**(6), 727–737 (2013)
4. Clearpath Robotics manufacturer site. <https://www.clearpathrobotics.com/>. Accessed 01 Sept 2019
5. ColRobot project site. <https://www.colrobot.eu/>. Accessed 01 Sept 2019
6. Tomatis, N.: BlueBotics: navigation for the clever robot [entrepreneur]. *IEEE Robot. Autom. Mag.* **18**(2), 14–16 (2011)
7. Sobreira, H., Moreira, A., Costa, P., Lima, J.: Mobile robot localization based on a security laser: an industry scene implementation. In: ROBOT 2015 - Second Iberian Robotics Conference, November 2015
8. Sobreira, H., Moreira, A., Costa, P., Lima, J.: Robust mobile robot localization based on security laser scanner. In: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions (2015)
9. Sobreira, H., Pinto, M., Moreira, A., Costa, P., Lima, J.: Robust robot localization based on the perfect match algorithm. In: 11th Portuguese Conference on Automatic Control CONTROLO 2014. Lecture Notes in Electrical Engineering (2014)
10. Sobreira, H., Moreira, A., Costa, P., Lima, J.: Robust mobile robot localization based on a security laser: an industry case study. *Ind. Robot: Int. J.* **43**(6), 596–606 (2016). <https://doi.org/10.1108/IR-01-2016-0026>
11. Sobreira, H., Costa, C., Sousa, I., Rocha, L., Lima, J., Farias, A., Costa, P., Moreira, A.: Map-matching algorithms for robot self-localization: a comparison between perfect match, iterative closest point and normal distributions transform. *J. Intell. Robot. Syst.* **93**(3–4), 533–546 (2018)
12. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: an efficient and accurate approach for robot self-localization. In: RoboCup Symposium, Osaka, Japan, 13–19 July 2005, pp. 142–153 (2005)