We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter 4

Occupancy Map Construction for Indoor Robot Navigation

Dora-Luz Almanza-Ojeda, Yazmín Gomar-Vera and
Mario-Alberto Ibarra-Manzano

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/64871

Abstract

Robot mobile navigation is a hard task that requires, essentially, avoiding static and dynamic objects. This chapter presents a strategy for constructing an occupancy map by proposing a probabilistic model of an ultrasonic sensor, during robot indoor navigation. A local map is initially constructed using the ultrasonic sensor mounted in the front of the robot. This map provides the position of the nearest obstacles in the scene useful for achieving the reactive navigation. The encoders allow computing the robot location in the initial local map. A first path for robot navigation based on the initial local map is estimated using the potential field strategy. As soon as the robot starts its trajectory in real indoor environments with obstacles, the sensor continuously detects and updates the occupancy map by the logsig strategy. A Gaussian function is used for modelling the ultrasonic sensor with the aim of reaching higher precision of the distance measured for each obstacle in the scene. Experiments on detecting, mapping and avoiding obstacles are performed using the mobile robotic platform DaNI 2.0 and the VxWorks system. The resulted occupancy grid is analysed and discussed at the end of this document.

Keywords: occupancy map, obstacle detection, path planning, robot mobile navigation, Gaussian model

1. Introduction

Nowadays, the artificial intelligent field has developed service task in robotic systems with the aim of providing help or comfort to humans. This is called service robotic [1], and it is

INTECH
open science | open minds

© 2016 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

supposed that the robot acts total or partially with autonomy. In particular, the service robotic tackles the problem of domestic robots. Such robots require highly autonomy and real-time processing for recognizing unknown environments. To do this, they use stereovision [2] and laser range finders [3, 4]. Furthermore, these robots must be able to autolocalizing and to navigate without human supervision, detect and locate obstacles and constructing its own map of the environment, using only their sensors and eventually a communication system with a central computer. The environment and workspace of the industrial robots are especially adapted for them and for performing special tasks; thus, these robots are programmed for knowing the environment in accordance with their physical dimensions. On the contrary, service robotics interacts with a changing and initially unknown environment.

Topologic and geometric maps could be constructed for perceiving robot environment. As proposed by Thrun et al. [5], a topologic map contains nodes and lines that joint these nodes representing the possible trajectories from one node to another. Furthermore, the author proposes the efficient mapping of the space and the low complexity as advantages, however, to recognize the place is difficult, depends of personal interpretation, and it could produce suboptimal trajectories. In contrast, the geometric maps numerically represent the coordinates and properties of the environment of the robot. These maps could efficiently represent big regions with few numeric parameters. Thus, the environment is descripted through geometric characteristics such as segments, corners, among others and their corresponding relationships (distance positions) [6].

On the other hand, the occupancy grid is a technique based on the discretization of the space in equal cells with a probability, which represents an occupied, empty or unknown area. This technique has been largely used due to it requires basic concepts for constructing, representing and updating and it allows to compute shorter trajectories. The main disadvantage of the occupancy grid is that requires the robot position.

The ultrasound is mechanical radiations with frequency higher to the audible range (>20 kHz) when these waves are reflected by an object in the environment. The ultrasonic sensors contain a piezoelectric transducer that is used as a transmitter and receptor for emitting and receiving the ultrasonic waves, respectively. This project proposes a strategy focused on ultrasonic sensors since this provides the distance from each obstacle in the environment to the sensor mounted on the robot. Recently, a self-configuring network of ultrasonic sensors has been proposed for tracking moving target [7], providing that these sensors provide an economical and basic solution to object detection in indoor environments.

In the development of novel strategies for proposing autonomous robotic systems, this work tackles with the problem of obstacle detection and avoiding in real time using an ultrasonic sensor embedded in a robot mobile during indoor navigation [8]. The environment is initially unknown; thus, first of all, the robot constructs an initial version of a two-dimensional (2D) occupancy map. Several experimental tests are performed for modelling the sensor error and for measuring the precision of measurements with the aim of assuring a reliable map.

http://dx.doi.org/10.5772/64871

2. Overall strategy for robot navigation

Figure 1 depicts the block diagram of the global strategy for constructing a local occupancy map of indoor environments. As the operating range of the ultrasonic sensor is known, this sets the size of the area covered by the sensor, which will be directly updated in the local map at each instant time *t*. This area establishes the size of an initial local map previously provided to the robot. On the other hand, before robot starts to move, the ultrasonic sensor is used for detecting objects in front of him. To do this, the ultrasonic sensor carries out a "sweep" in the range −90° to 90° with respect to X axis of the robot (see **Figure 2a**). The local map also uses the initial robot location, due to location of the objects are given with respect to the ultrasonic sensor. The path-planning module computes an initial path for robot navigation. The potential field's technique is used to plan the best trajectory for the robot. In real time, robot odometric location is obtained from the encoders for updating the map and continuously constructing the global map of the robot scene. Both global and local maps store the probability of occupancy around the robot during navigation.

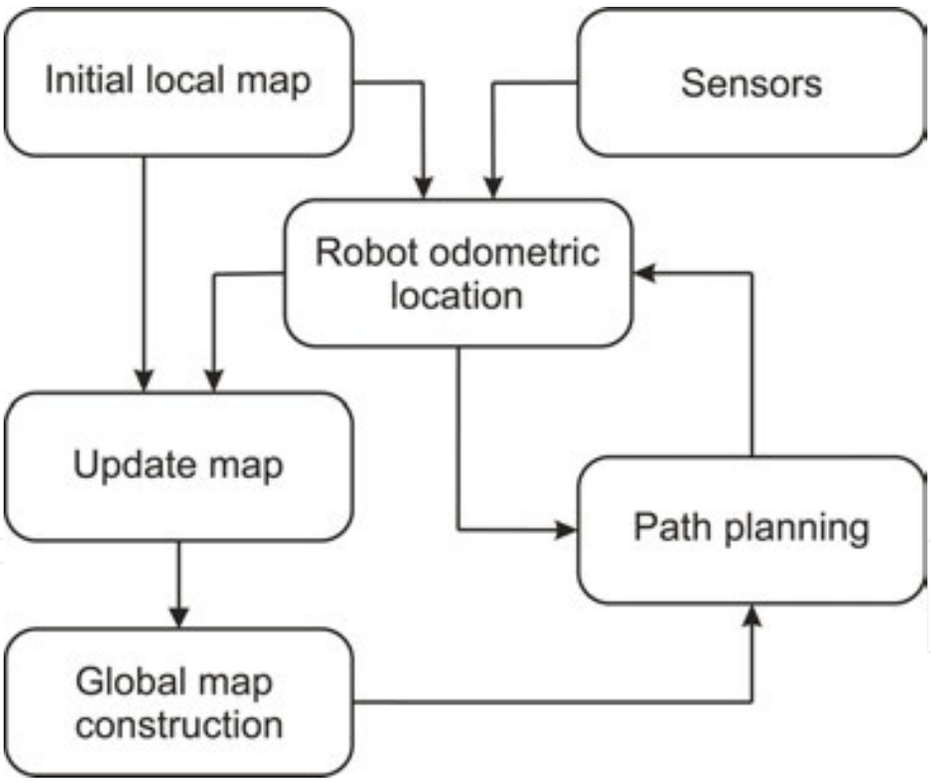


Figure 1. Global strategy for constructing an occupancy map during indoor navigation.

Every update of the global map is stored in a file internally on the robot. The map construction consists in dividing the environment in small uniform cells, which will be labelled as occupied or free in accordance with the ultrasonic measures. An intensive calibration strategy is required with the aim of obtaining an accurate digital representation of the real scene. The robot

navigates from a predefined initial position to a goal position; then, the algorithm ends when the robot reaches such predefined goal position.

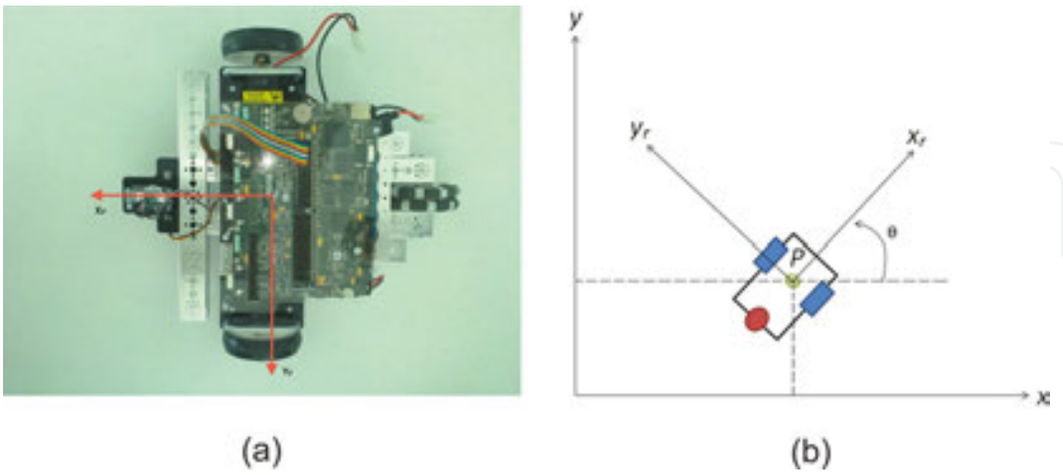


Figure 2. (a) Reference axis of the mobile robot DaNI 2.0. (b) Graphical representation of the robot in the local and global reference axis, P represents the reference point of the position.

3. Robot model description

The mobile robot used in this project is shown in **Figure 2a**. This is a robotic platform called NI LabVIEW robotics Starter Kit® [9], also known as DaNI 2.0, developed by National Instruments company® (NI). This mobile robot was designed to develop and run algorithms in real time for autonomous system applications. The components of the mobile robot DaNI 2.0 are essentially: two DC motors, two encoders and a reconfigurable card sbRIO-9632 (Single Board Reconfigurable I/O), an ultrasonic sensor mounted on a servomotor for providing to sensor rotational motion. **Figure 2b** illustrates a graphical representation of the robot, with the global and local reference axis of the robot. The X and Y axes are defined arbitrarily in the plane as the reference of the global coordinates.

The kinematic model of the robot DaNI 2.0 consists in a differential configuration: each wheel of the robot is connected to a DC motor, which provides the traction force and a stabilization wheel for balancing the robot. The basic model for representing the robot position considers the robot as a single point in the space. Thus, the robot position is specified by choosing a point *P* in the robot chassis as a reference; usually, this point is the centre of the wheel axis. In addition, the point *P* represents the origin of the robot axis X_r and Y_r indicating the local reference of the robot position, see **Figure 2a**.

In particular, the sbRIO-9632 card is a heterogeneous-embedded platform, developed by NI®, which contains a real-time processor and serves as main control unit of the robot. Besides, this platform includes a Field Programming Gate Array (FPGA) Xilinx Spartan-3, which is a reconfigurable device that executes programmed tasks in real time, that is, the active response

of the system to external events. This real-time execution uses a low-level programming to perform tasks, such as motor control, signal acquisition by means of digital or analogue inputs and monitoring digital and analogue inputs/outputs, among others. However, a higher level of programming is possible through the NI LabVIEW® robotics software, which is a graphical language. Programming languages such as C, C++ or Java could be also used.

3.1. Robot odometric localization

During navigation, the robot needs to know its position and orientation with respect to its local and global axis of reference. To do this, the most common method used is based on geometric equations providing an estimation of the robot location by combining information obtained from the encoders on the wheels and from the propulsion components. This method is known as odometric estimation [10], and it is commonly used due to it only employs the kinematic model of the robot without including forces or torques in the mechanism. The main constraint of this method is that, a small error at the beginning of the estimation increases with the time, due to it is accumulated. Another strategy for avoiding this incremental error is to correct the robot position at regular times of movement, using landmarks [11]. Nevertheless, the cost of installation of the landmarks could be considerably high. For this reason, the odometric location strategy is commonly used providing enough results in short trajectories at low cost.

In order to compute the robot position $P(X, Y)$ in the global coordinate system, it is necessary to know the rotational angle between the local and global coordinate axis, given by θ and the origin coordinates of the robot local axis. The geometric equations for computing the point P position in global coordinates are as follows:

$$\begin{cases} X = X_r \cos(\theta) \\ Y = Y_r \sin(\theta) \end{cases} \tag{1}$$

The robot motion d_r is computed over the time by considering the robot geometry and its angular velocity on the wheels, using the following equation:

$$d_r = \frac{2\pi r}{rev} \varphi_\omega t \tag{2}$$

where r is the wheel radius (1.16 dm for the DaNI 2.0 robot), φ_ω the angular velocity of the wheel and t is the time.

In accordance with the robot geometry and the wheel type, the movement of the robot X_r is only in one direction (X direction is used). Furthermore, in the case of one wheel is keeping fix to the floor, while the second wheel moves with an angular velocity; then, the robot will draw a circle around the fixed wheel with a radius of $2l$; being l the distance from the point P in the chassis to the wheel (1.778 dm for DaNI 2.0 robot). This movement only affects θ angle:

$$\theta = \frac{r\varphi_{\omega}t}{2l} \tag{3}$$

In order to estimate the location errors due to the odometry, a square trajectory was implemented on the robot. It was found that such errors in the trajectory are mainly due to the errors in the turns, performed to 86° approximately but expected to 90°. Besides, small changes in the wheels' trajectory are not registered by the encoders, these changes are minimal, and however, they produce a notable final error.

The odometric errors exist essentially due to the robot construction, that is, there is a small difference between the diameters of the wheels. Furthermore, the finite resolution of the encoders and the irregularities in the floor where the trajectory is performed avoid an ideal execution of the trajectory.

3.2. Ultrasonic sensors

As it was mentioned in the introduction section, the ultrasonic sensors consist of one transmitter and receptor of the sound. Once the ultrasonic wave has been sent, when such wave found an object a signal, this is reflected as an echo and can be detected by the same transmitter, which acting also like a receptor. In general, the applications of the ultrasonic sensors are based on estimate the lapse of time between such emission and reception of the ultrasonic waves. This lapse of time is known as time of flight (ToF), the corresponding distance to the object that reflected the wave is estimated by means of:

$$d = \frac{1}{2}v \cdot t_f \tag{4}$$

where v represents the velocity of sound and t_f the time of flight.

On the other hand, one particular problem with the ultrasonic sensors is the mirror reflection. This reflection happens mainly in the corners and it is occasioned for several reflections of the ultrasonic waves before to return to the sensor (see **Figure 3**). As a consequence of this phenomenon, some objects of the environment of small size or orientation cannot be detected by the sensor, or in some cases, they are detected farther than really they are. Thus, only the readings taken when the ultrasonic wave impacts perpendicular to the surface will be taken as correct measures.

The cone of sensibility, also known as acoustic sensor aperture, introduces incertitude in the position and distance of the reflected object if the robot is in motion. This is illustrated in **Figure 4**: if one object is detected in the cone of sensibility, therefore, the measured distance will correspond to an object "in front" of the sensor, even if the object is located with an orientation with respect to the robot.

http://dx.doi.org/10.5772/64871

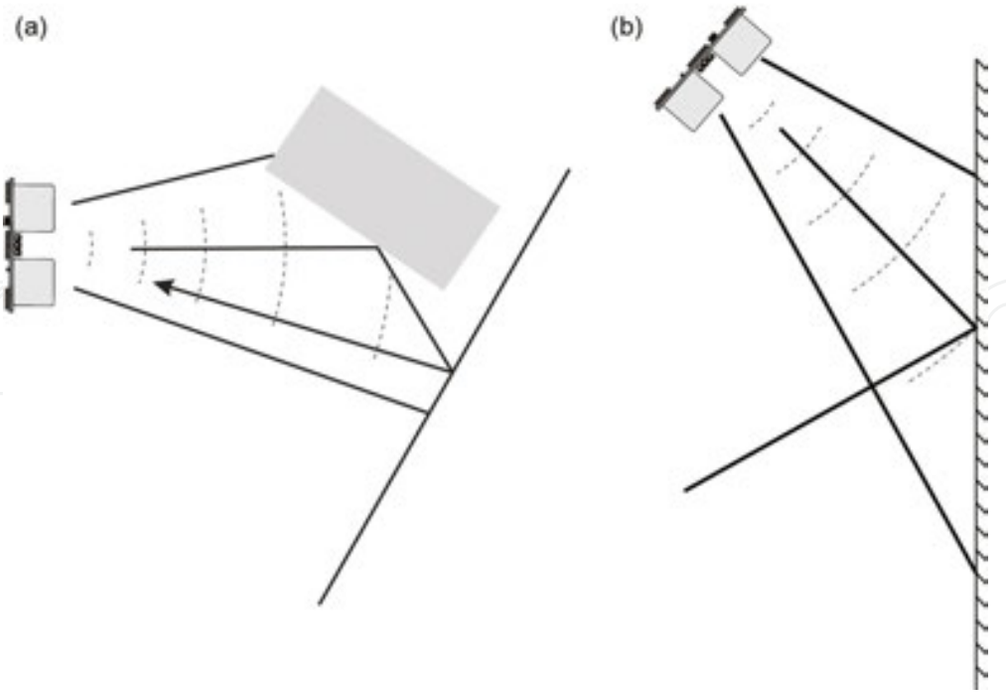


Figure 3. Mirror reflection of the ultrasonic waves. (a) Extended trajectory of the ultrasonic wave. (b) Ultrasonic echo that does not return to sensor.

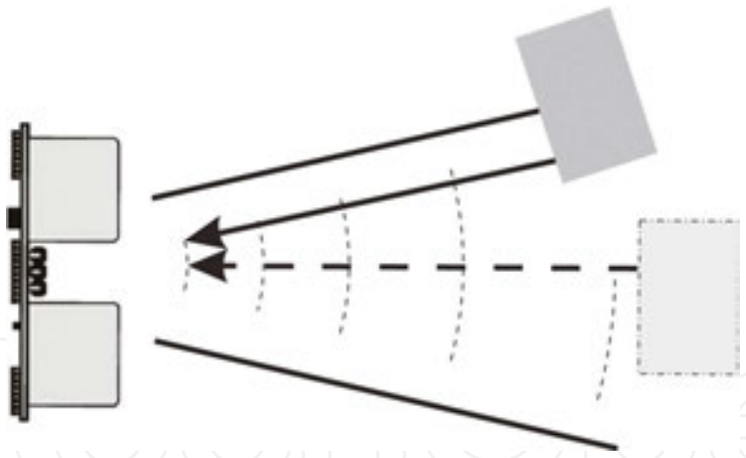


Figure 4. Main effect of the acoustic aperture of the ultrasonic sensor that produces incertitude in the object position, positioning the object in an unreal location.

3.3. Gaussian model of the ultrasonic sensor

The experimental test for modelling the sensor was performed at different sampling times with the aim of considering the effect of the robot motion in the measurements. The sampling times used were 60, 80, 100, 200 and 400 ms, in the range of −65° to 65° with intervals of 5° at distances of 6, 8 and 10 dm from the wall. **Figure 5** illustrates the average of these different test performed using a sampling time of 80 ms. The chart (a) shows a stable range of measurements among

−20° to 40° confirmed by the standard deviation shown in chart (b). Note that different sensor positions (6, 8 and 10 dm) do not affect the measurements. Similar results have been obtained for a sampling time of 100 ms; however, as this project will be performed in real time, 80 ms was chosen due to it represents the best trade-off between stability and time of detection.

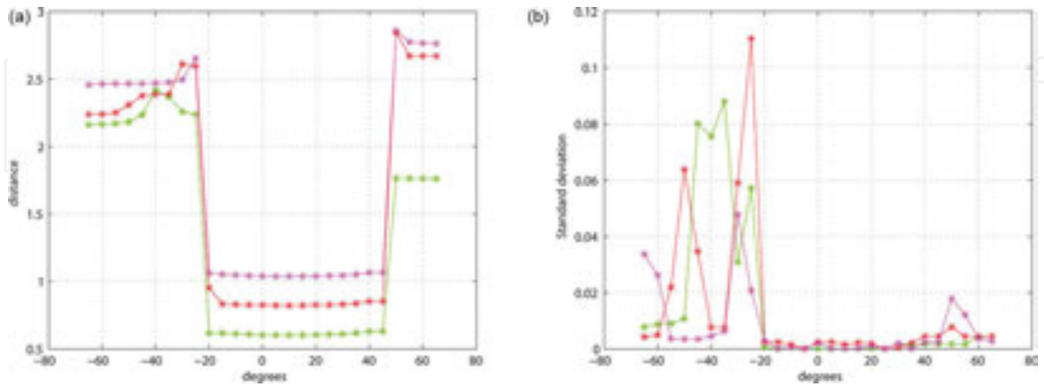


Figure 5. Charts of the measurement averages. (a) The sampling time used is 80 ms, for three distances 6, 8 and 10 dm. (b) The standard deviation of measurements performed in chart (a), note the range of stability is among −20° to 40°.

In order to reduce the measurement error of the ultrasonic sensor and to validate that the detected object is inside a specific region, it is used a probabilistic technique based on a Gaussian function. The Gaussian model implies to know the errors due to the distance and angle detection [12]. This model considers the measured distance, denoted here as d, and its uncertainty (σ_d), and orientation angle θ and its uncertainty (σ_θ). Therefore, the real measure is in the range of $d \pm \sigma_d$, and the orientation $\theta \pm \sigma_\theta$ is denoted by:

$$P(z|d,\theta) = \frac{1}{2\pi\sigma_d\sigma_\theta} e^{\left(\frac{-(d-z)^2}{2\sigma_d^2} + \frac{\theta^2}{\sigma_\theta^2}\right)} \tag{5}$$

where z is the variable in the workspace of the ultrasonic sensor measurements. This equation represents the probability that the object be in the position measured by the sensor and uses two standard deviations, range and angle (see **Figure 6a**). The error measured on the angle for one object located at 10 dm is 0.12 dm that represents an error of ±0.7°. This effect is small; therefore, it will be ignored for constructing the occupancy map, considering only the 1D model based on distance only (see Eq. 5). **Figure 6b** depicts in blue the 1D model of the ultrasonic sensor considering only the distance and the uncertainty (σ_d). Using a confiability value of 0.8, the resulted plot is shown in green. Once the object has been detected, the environment behind him is unknown. Thus, the probability of such cells is considered 0.5 because the cell value cannot be known (red plot in **Figure 6b**). Furthermore, this plot represents the probability of the object be in the distance describe in the x axis. This result will be used for occupancy map construction.

http://dx.doi.org/10.5772/64871

$$P(d|z) = \frac{1}{\sqrt{2\pi\sigma}} e^{\left(\frac{-(d-z)^2}{2\sigma^2}\right)} \tag{6}$$

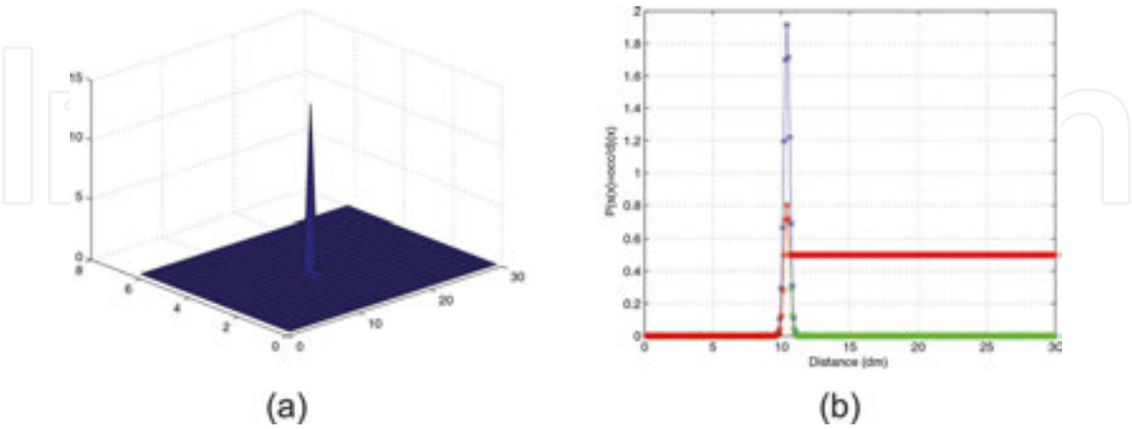


Figure 6. Gaussian model of the ultrasonic sensor. (a) Object position in coordinates (d, θ) for a 2D probability model. (b) Probability model of object position detected by ultrasonic sensor in 1D.

4. Global map construction

The occupancy maps are a probabilistic technique based on small cells that divide the surrounding space of the indoor or outdoor environments. The probability of one cell is occupied which is estimated using robot sensors. Each cell in the map represents the information contained in the physical space in front of the sensors used to measure the environment. The values in the cell describe the following situations:

$$\begin{cases} <0.5 & \text{free cell} \\ =0.5 & \text{unknown} \\ >0.5 & \text{occupied cell} \end{cases} \tag{7}$$

4.1. Static occupancy map

The initial local map uses the initial position of the robot, which is always knowing as navigation is performed only in indoor environments for constructing a static occupancy map considering that the robot is not moving in the environment. The ultrasonic sensor in front of the robot rotates in the range of −90° to 90° with respect to the x axis of the robot; nevertheless, the real range used was in the range of −20° to 40° in order to acquire more stable and accurate measures. Another advantage is that a reduced amount of measurements avoiding synchronization problems and allowing the real-time execution. The rest of the angle range will be covered as the robot navigates in the environment.

In this work, an initial local map is constructed since a static position of the robot in the environment which is represented by a grid as illustrated in **Figure 7b**. This map indicates the starting point of the robot trajectory and the corresponding state of the surrounding environment in accordance with Eq. (7). Each cell of the map represents 1 dm^2 of the environment minimizing the location errors due to the odometry of the robot. Note that, the red cells in the front and in both sides of the robot represent the obstacles in the scene (probability of the cells are higher to 0.5). This map is updated based on the log-odds algorithm for mapping the environment in a global map.

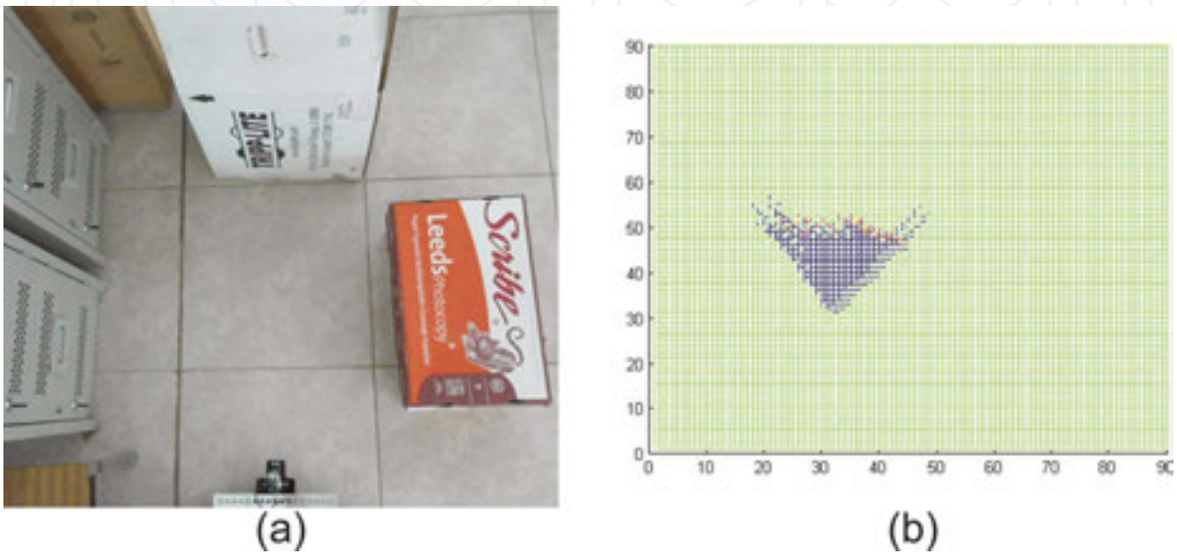


Figure 7. Initial local map. (a) Indoor scene with obstacle, (b) blue cells represent free space, red cells represent obstacles and green cells are unknown space.

4.2. Dynamic occupancy map

Figure 8 illustrates the block diagram for constructing the dynamic occupancy map. The measures obtained from the encoders in order to locate the robot in the global reference system. Once the current robot location in the environment is known, a homogenous transformation is carried out with the aim of updating the cells covered by the ultrasonic sensor based on the Eq. (7) and finally assigning them in the global map.

Note that, in the odometric localization block of **Figure 8**, the local reference axis of the robot has been rotated and angle β with respect to global axis. Therefore, one point (X_1, Y_1) in the local robot axis oriented with an angle α will be located in the global reference axis by means of:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & O_X \\ \sin\alpha & \cos\alpha & O_Y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} \tag{8}$$

http://dx.doi.org/10.5772/64871

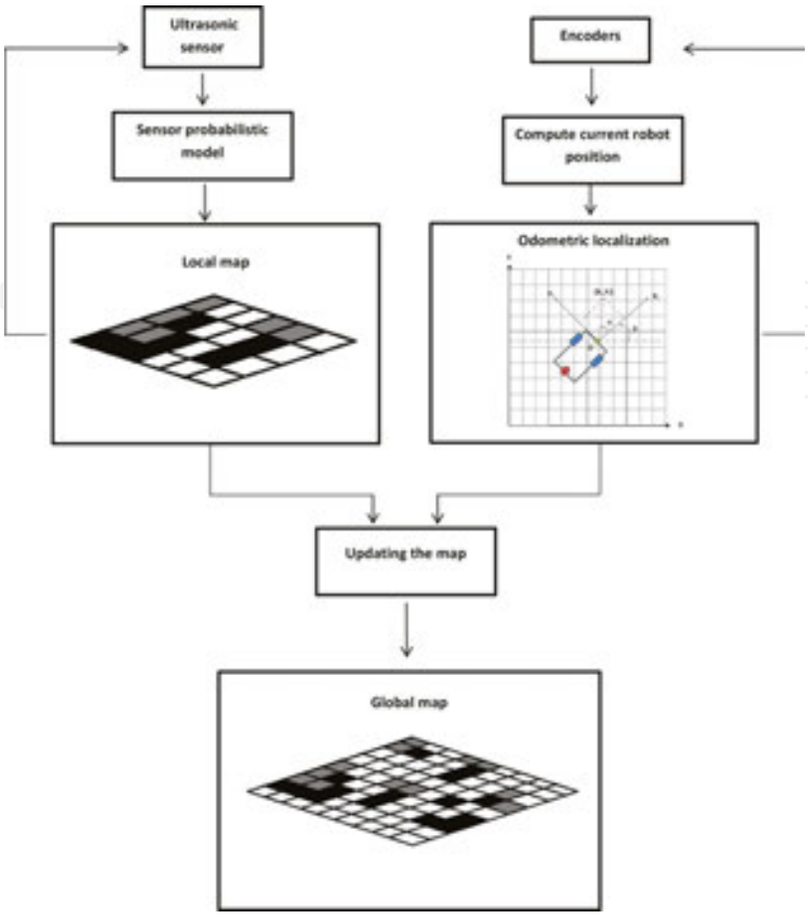


Figure 8. Local and global reference axis for constructing the dynamic occupancy map.

4.3. Updating the global map

One commonly used technique for updating the occupancy map is the Bayesian strategy, described in Eq. (10), which defines the probability P of the cell state $s(C_i)$ be occupied, given the distance (d) measured by the sensor at the time $t+1$.

$$P\left[s\left(C_i\right)=OCC\big|_{d_{t+1}}\right]=\frac{P\left[d_{t+1}|s\left(C_i\right)=OCC\right]P\left[s\left(C_i\right)=OCC|d_t\right]}{\sum_{s(C_i)}P\left[d_{t+1}|s\left(C_i\right)\right]P[s(C_i)|d_t]} \tag{9}$$

Each lecture provides partial information of the environment; therefore, in order to establish the state of each cell, an updated equation is used for combining the prior and the current probabilities of the cell, yielding:

$$P\left(C_i\right)=P(C_i)_{t-1}+P(C_i)_t \tag{10}$$

4.4. The log-odds algorithm

Eq. (10) could give numerical instabilities for probabilities closer to 0 or 1. To overcome this problem, Thrun [13] propose that the status "occupied" for each cell *i* at instant time *t* can be modelled as the logarithm of an occupied cell, divided by the probability of the cell be empty and represented by:

$$l_{t,i} = \log \frac{P\left[\left(C_i\right) | d\right]}{1 - P\left[\left(C_i\right) | d\right]} \tag{11}$$

The probability could be obtained, easily:

$$P\left(C_i\right) = 1 - \frac{1}{1 - e^{l_{t,i}}} \tag{12}$$

Besides, it is considered that the prior value of the occupied cells l_0 takes a constant value given by:

$$l_0 = \log \frac{P(C_i = 1)}{P(C_i = 0)} = \log \frac{P(C_i)}{1 - P(C_i)} \tag{13}$$

The algorithm for updating the occupancy map is described as follows:

Algorithm 1. Updating the occupancy map

Inputs: $\{l_{t-1,i}, r, z\}$

for each cell C_i do

if C_i is in the vision field of the sensor then

$l_{t,i} = l_{t-1,i} + l_{sensor_model} - l_0$

else

$l_{t,i} = l_{t-1,i}$

end if

end for

5. Path planning

A safe and coherent navigation of the robot in his workspace requires a path planning technique and an obstacle avoidance strategy. In this project, the path planning technique used

http://dx.doi.org/10.5772/64871

is the potential fields which basically consist in the computation of the attraction forces produced by the goal position and the repulsion forces caused by the closer objects in the scenario. Thus, an artificial potential field guides the robot to goal position, while the obstacle avoidance method allows a safe navigation of the robot.

The potential field acting on the robot is given by the attraction field towards the goal and the repulsion field produced by the obstacles, that is:

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q) \tag{14}$$

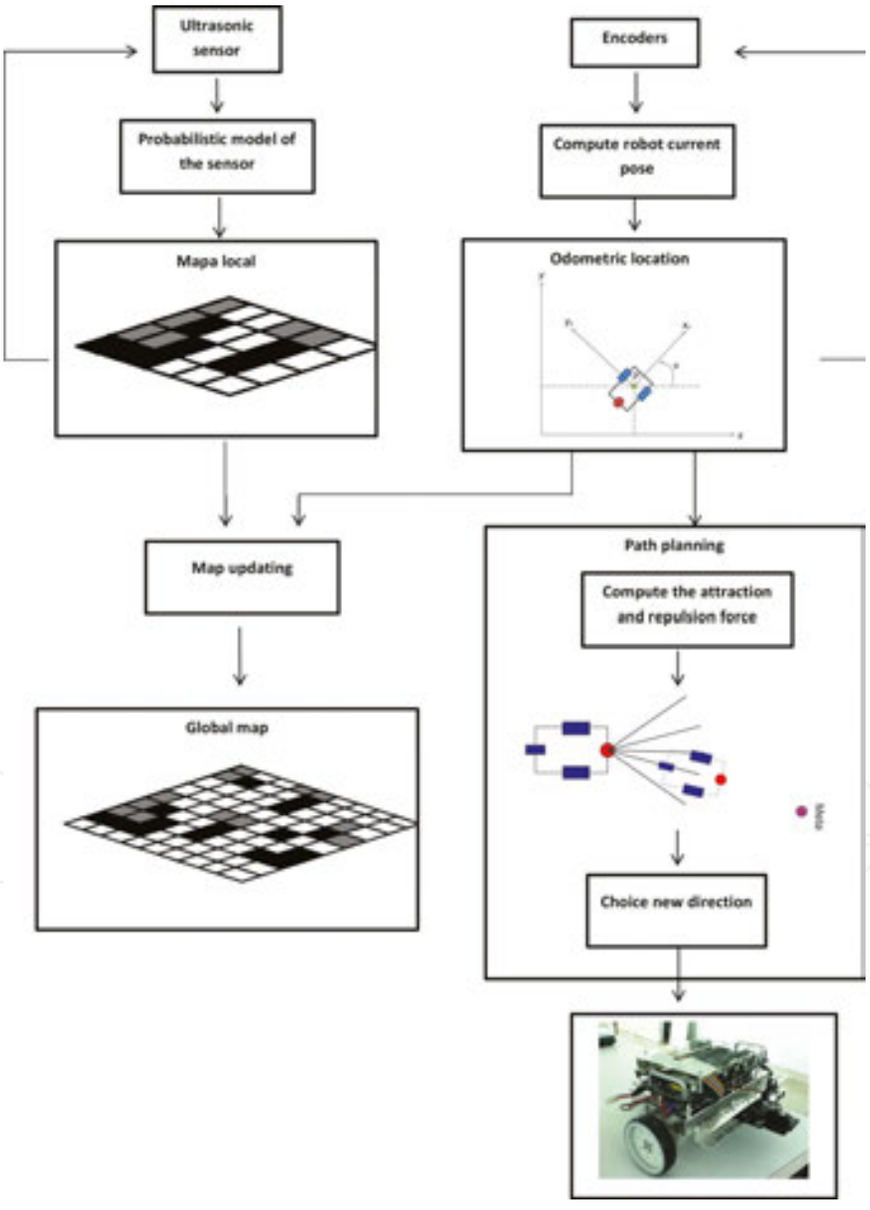


Figure 9. Block diagram of the path planning algorithm.

In the same way, the forces could be separated in attraction and repulsion forces yielding:

$$F(q) = F_{\text{att}}(q) - U_{\text{rep}}(q) \tag{15}$$

Furthermore, the attraction force could be described as:

$$F_{\text{att}}(q) = -k_{\text{att}} \cdot \left| \left(q - q_{\text{goal}} \right) \right| \tag{16}$$

where $\rho_{goal}(q)$ is the Euclidean distance between $\| q - q_{goal} \|$, and the repulsion force could be described as:

$$F_{\text{rep}}(q) = \begin{cases} \frac{1}{2} k_{\text{rep}} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \frac{q - q_{\text{obstacle}}}{\rho(q)} & si\ \rho(q) \le \rho_0 \\ 0 & si\ \rho(q) \ge \rho_0 \end{cases} \tag{17}$$

where k_{att} and k_{rep} are positive scalar factors, $\rho(q)$ is the minimal distance from q to the object and ρ_0 is the distance in which an object influences in the path.

The general strategy for programming the path planning algorithm is illustrated in **Figure 9**. If the current robot position is known, then the resultant force is estimated considering all the new directions that the robot can take. Such directions are chosen based on the angle of observation of the ultrasonic sensor and the probable position of the robot if he walks in that specific direction. Once each force has been computed, the smaller is selected due to this force moves the robot closer to the goal. Once the new direction has been selected, the motors are turned on to follow this direction. The algorithm ends when the goal has been reached by the robot.

6. Experimental results

Several tests were performed to validate the path planning algorithm. In the tests, the mobile robot DaNI 2.0 navigates in an indoor environment with obstacles located between the starting and goal position. Both starting and goal positions are the same during the experiments and only the obstacle locations change. The graphical representation of the occupancy map is labelled as: green colour represents the unknown environment, blue cells represent free path (empty cells) and red cells represent obstacles (occupied cells).

The first experimental test is performed in an environment without obstacles; the results are illustrated in **Figure 10**. The odd rows of this figure show different instant times of the whole sequence performed by the robot; in particular, the first picture shows the starting position,

http://dx.doi.org/10.5772/64871

and the ninth picture shows the goal position of the robot. Even rows depict the occupancy map constructed at that time. Note how effectively the obstacles detected are located in the borders of the region navigated by the robot, that is, any obstacle is detected in the middle of the scene as is expected.

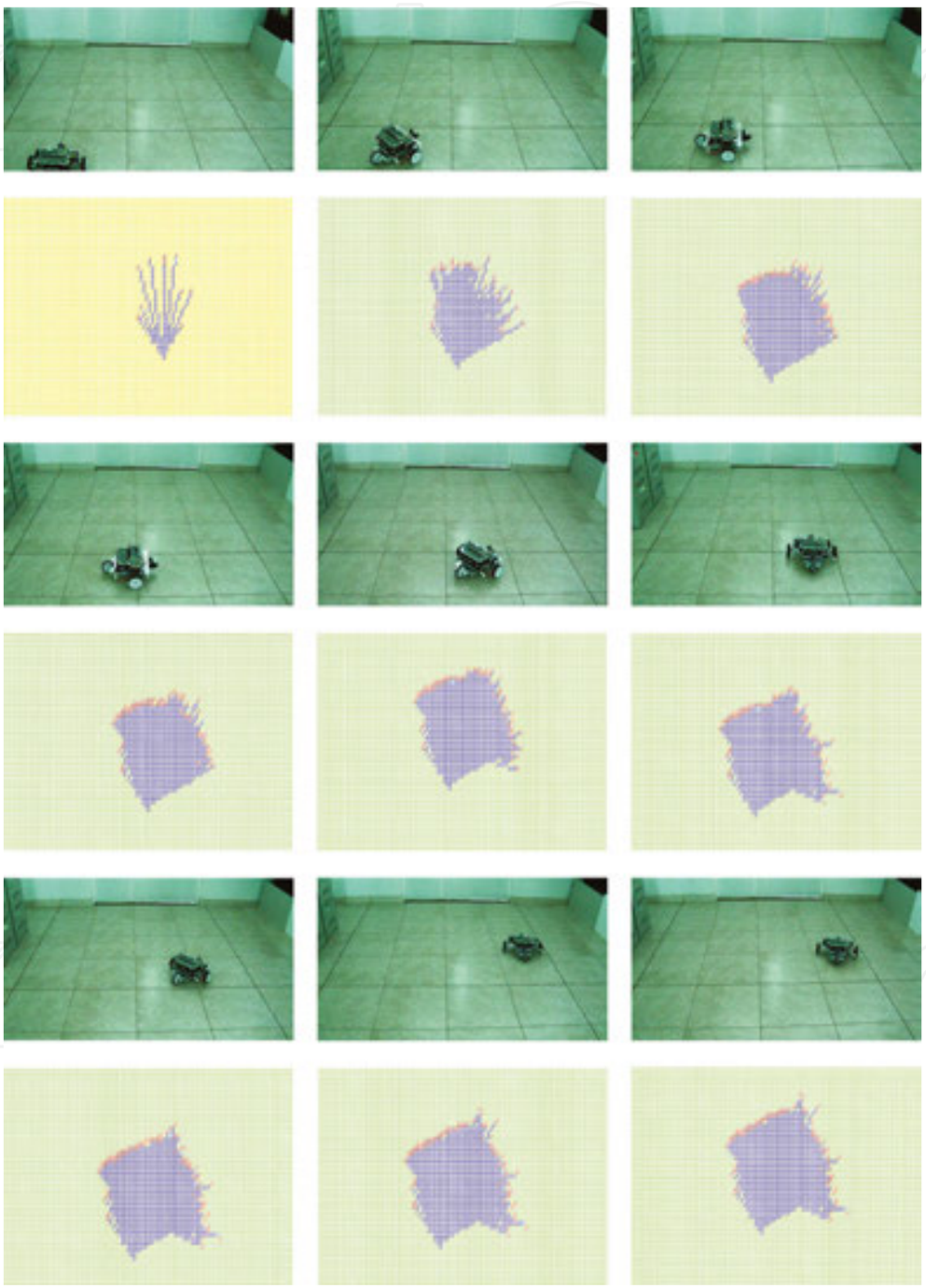


Figure 10. Test 1 of the path planning algorithm. The navigation is performed without obstacles. The graphical representation of the occupancy map is updating using the measurement obtained from the ultrasonic sensor, during robot navigation in real time.

The second experimental test is performed in an environment with two obstacles; the results are illustrated in **Figure 11**. In this case, the first picture shows the starting position, and the ninth picture shows the goal position of the robot. Note that the final occupancy map constructed is smaller than the map of the test 1. The last is due to obstacles are detected therefore included as red cells; however, behind the object, even if there are a free space, this area is not reached for the ultrasonic sensor.

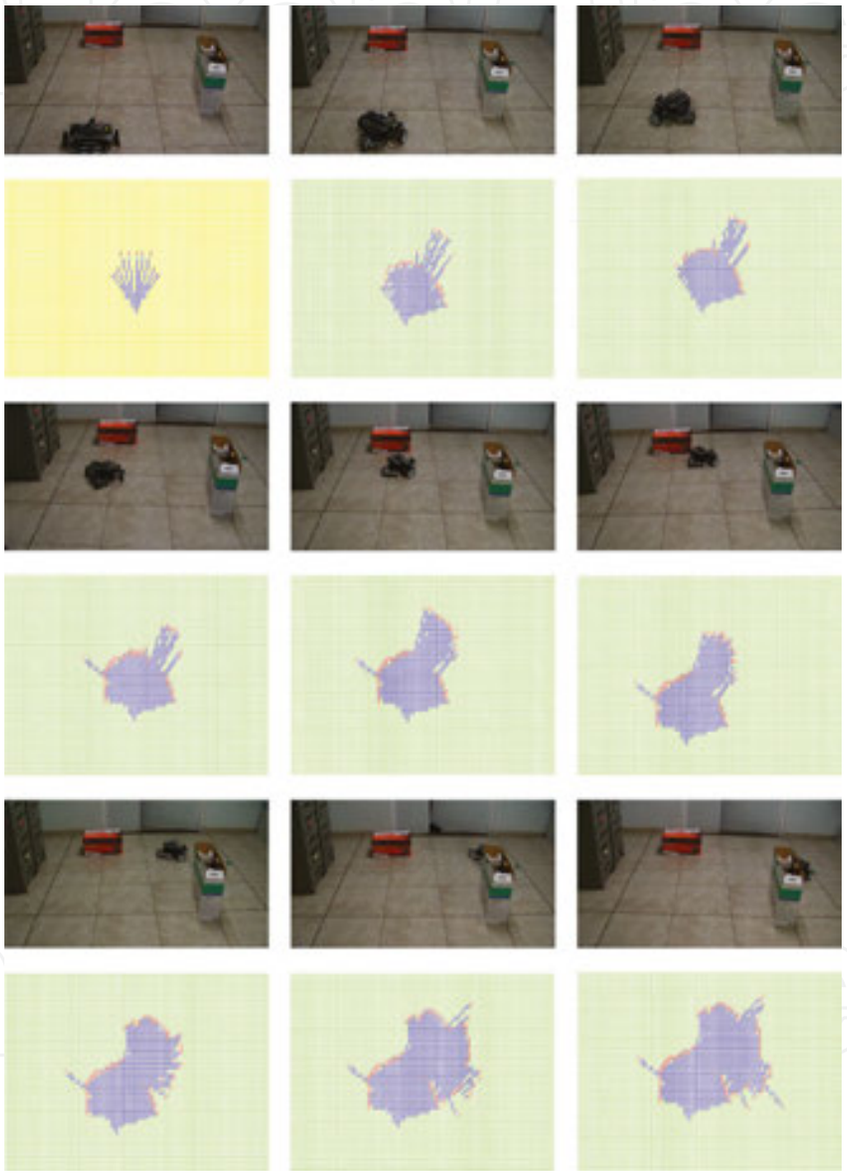


Figure 11. Test 2 of the path planning algorithm. The navigation is performed with two obstacles in the environment. The graphical representation of the occupancy map is updating using the measurements obtained from the ultrasonic sensor, during robot navigation in real time.

The results of the last experimental test showed here are illustrated in **Figure 12**. As in the test 2, the occupancy map is small with respect to the real environment due to zones behind objects cannot be not established by the robot.

http://dx.doi.org/10.5772/64871

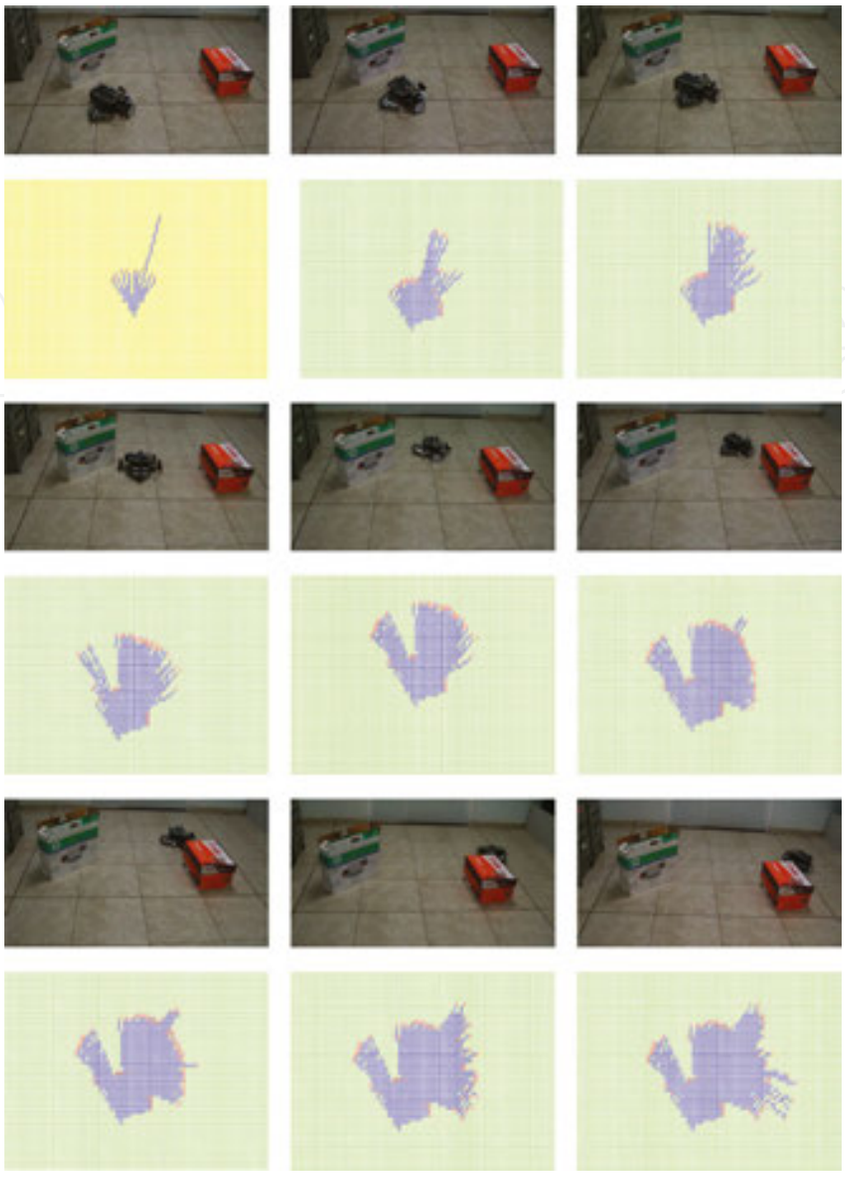


Figure 12. Test 3 of the path planning algorithm. The navigation is performed with two obstacles in the environment at different position with respect to test 2. The graphical representation of the occupancy map is updating using the measurements obtained from the ultrasonic sensor, during robot navigation in real time.

7. Conclusions and perspectives

The construction of occupancy map using ultrasonic sensors is easily perturbed by environmental noise. To overcome this constraint, it is used a Gaussian model of the sensor, providing higher precision of the distance measured for each obstacle in the scene. In addition, another typical error found in the measurements performed by ultrasonic sensors is related with the cone produced at the time of sending the ultrasonic signal. In this project, it was found that this error is less than 1°; therefore, it could be ignored. Furthermore, the error produced by the

irregularities in the floor is reduced by using the proportional integral-derivative (PID) control of the robot. Updating the global map requires the synchronization between the ultrasonic measures and the robot location in the environment.

The occupancy map of a real indoor environment is constructed by an ultrasonic sensor and a mobile robot. The logsig strategy allows a safe indoor navigation of our robot by establishing a probability of occupancy to each cell in the map avoiding a collision of the robot with an obstacle. The global strategy is performed in real time. The time employed for the robot for constructing the map depends of the number of obstacle in the scene; however, a real-time execution of the platform is assured by using the VxWorks platform of the robot DaNI 2.0.

Future works are focused on multiple robot navigation on dynamic environments. To do this, a proposed approach consists in sending the local map constructed for one robot to a master terminal by IP connection in order to incrementally construct the global map by adding local maps of the environment.

Author details

Dora-Luz Almanza-Ojeda*, Yazmín Gomar-Vera and Mario-Alberto Ibarra-Manzano

*Address all correspondence to: luzdora@ieee.org

Electronics Engineering Department, DICIS, University of Guanajuato, Salamanca, Guanajuato, México

References

[1] Somolinos Sánchez J. A., editor. Advanced robotics and computer vision, 1st ed., Spain : Ediciones de la Universidad de la Mancha ; 2002. pp. 285 p.

[2] Tavera-Vaca C.-A., Almanza-Ojeda D.-L., Ibarra-Manzano M.-A. Analysis of the efficiency of the census transform algorithm implemented on FPGA. Microprocessors and Microsystem. 2015;39(7):494–503. doi:10.1016/j.micro.2015.08.002

[3] Ventura R., Ahmad A. Towards Optimal Robot Navigation in Domestic Spaces. In: Reinaldo A. C., Bianchi, H. Levent Akin, Subramanian Ramamoorthy, Komei Sugiura, editors. RoboCup 2014: Robot World Cup XVIII. 1st ed. Switzerland: Springer; 2015. p. 318–331. doi:10.1007/978-3-319-18615-3_26

[4] Martínez M. A., Martínez J. L., Morales J. Motion Detection from Mobile Robots with Fuzzy Threshold Selection in Consecutive 2D Laser Scans. Electronics, 2015 ; 4(1) : 82-93. Doi : 10.3390/electronics40110082

[5] Thrun S., Burgard W., Fox D. Probabilistic Robotics. 1st ed. Cambridge, MA: The MIT Press; 2005. pp. 672 p.

[6] Leonard J. J., Durrant-Whyte H. F. Mobile robot localization by tracking geometric beacons. IEEE Transactions on Robotics and Automation. 1991;7(3):376–382. doi: 10.1109/70.88147

[7] Teixido M., Palleja T., Font D., Tresanchez M., Moreno J., Palacin J. Two-Dimensional radial laser scanning for circular marker detection and external mobile robot tracking. Sensors. 2012;12(12):16482–16497. doi:10.3390/s121216482

[8] Almanza Ojeda D. L., Gomar Vera Y., Ibarra Manzano M. A. Obstacle Detection and avoidance by a mobile robot using probabilistic models. IEEE Latin America Transactions. 2015;13(1):69–75. doi:10.1109/TLA.2015.7040630

[9] National Instruments. LabVIEW Robotics [Internet]. 05/01/2016 [Updated: 05/01/2016]. Available from: http://sine.ni.com/psp/app/doc/p/id/psp-912/lang/es [Accessed: 05/15/2016]

[10] Klarer P. 2D Navigation for Wheeled Vehicles. 1st ed. Albuquerque: Sandia National Laboratories; 1998.

[11] Borenstein J., Feng L., Everett. Where am I? Sensors and Methods for Autonomous Mobile Robot Localization. 1st ed. Michigan: The University of Michigan; 1994.

[12] Kuc R., Barshan B. Navigating vehicles through an unstructured environment with sonar. In: IEEE, editor. IEEE International Conference Robotics and Automation; 24–29 May 1989; Scottsdale, AZ. IEEE; 1989. p. 1422–1426. doi:10.1109/ROBOT.1989.100178

[13] Thrun S. Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence. 1998;99(1):21–71. doi:10.1016/S0004-3702(97)00078-7

IntechOpen

IntechOpen

IntechOpen