Universitat de Lleida

Escola Politècnica Superior

Enginyeria en Informàtica

Sistemes Informàtics (Treball de final de carrera)

Optimization of Floor Cleaning Coverage Performance of a Random Path-Planning Mobile Robot

Autor: Marcel Tresanchez Ribes

Director: Jordi Palacín Roca

Juny 2007

Contents

1 Objective of this work

This work has been developed jointly with the Robotics Group of the University of Lleida where the expertise is focused on applications of mobile robots; mainly for floor cleaning.

The main objectives of this work are:

- Obtain an optimal random path-planning algorithm for a cleaning mobile robot.
- Evaluate some methods to estimate the area of the scenario to be cleaned (using only encoders and collision detectors).
- Develop a method to estimate the evolution of the area cleaned by the robot.

The main constrain of this work is that all the methods and procedures must be applied to a very simple mobile robot with very few sensors to guarantee its industrial interest.

2 Introduction

Floor cleaning is a noncreative human task that can be automated using mobile robots. Coverage algorithms based on genetic algorithms [3, 4], neural networks [5, 6], exact cell decomposition [15], spanning trees [7], spiral filling paths [8, 9, 10], etc, are highly sensitive to the internal [16, 17] and external [21, 22, 23] sensors of the mobile robot and the previous or acquired information of the area where the robot operates [1, 2]. Moreover, the typical domestic cleaning scenario becomes unstructured and unknown by the furniture, decoration things and the typical human disorder habits. Then the complete-coverage problem becomes more and more complex [11-15] and high quality sensors as ultrasonic or laser are needed to obtain a map [18-20] of the cleaning scenario.

However, the high cost of the sensors precludes the commercial development of efficient cleaning robots. At this moment, most of the commercially available mobile robots are based on random path-planning algorithms [24, 25] with very few sensors as magnetic encoders and contact or non-contact collision detection. Nevertheless, even a cleaning with a random path-planning algorithm can be optimized because the cleaning area has an exponential evolution that can be modeled using one exponential [24, 25]. In [25] the authors have demonstrated that the time constant and amplitude of the exponential can be estimated if the size of the area is known. To this end, the main objectives of this work are to obtain an optimal random path-planning algorithm, evaluate some methods to estimate the area of the scenario to be cleaned (using only encoders and collision detectors) and a method to estimate the evolution of the area cleaned by the robot. The main constrain of this work is that all the methods and procedures must be applied to a very simple mobile robot with very few sensors to guarantee its industrial interest.

The outline of the paper is as follows: first, Section 2 describes the simulator used and the real mobile robot modeled in the simulations. Section 3 proposes an optimal random path-planning algorithm based on the statistical analysis of some simulation results. Section 4 proposes three methods to estimate the size of an ideal cleaning scenario and four methods to estimate the area of real rooms with furniture. Section 5 proposes a method to estimate the evolution of the area cleaned by the robot. Finally, Section 6 presents some conclusions.

3 Simulating Floor Cleaning Mobile Robot

3.1 Mobile Robot Simulator

A complete mobile robot simulator called Simrobot was developed for MatLab. The simulator can be programmed with a detailed model of a real mobile robot including size, wheel diameter, speed of the motors, error sources, size and position of the cleaning device, collision sensors and non-contact infrared and ultrasonic sensors. The simulator can also include a basic or detailed description of a given scenario. The main objective of the simulator is to reproduce the dynamic behavior of a real mobile robot and trace some robot parameters as area cleaned. Figure 1 show an example of control of a mobile robot through the simulator. A movement is proposed and the real movement, the status of the collision sensors and the virtual time spend by the robot in the movement is returned.

```
while (t < 3*60)
        [distance, collision, t] = simrobot('forward',Inf);
        [distance, collision, t] = simrobot('backward',10);
        [angle, collision, t] = simrobot('turn',360*(rand(1)-0.5));
end
```

Figure 1. Basic random path-planning control of a mobile robot: distances in cm, angles in degrees and time in s.

3.2 Mobile Robot Simulated

The mobile robot simulated is an evolution of [23] called Robonet (Figure 2). It has a diameter of 28.6 cm, a height of 10 cm and a total weight of 2 kg batteries included. It can run at a maximum speed of 1 km/h but the nominal speed is 0.3 km/h. Robonet includes magnetic encoders in the motors, one two-axis silicon accelerometer and several ultrasonic sensors. However, only the collision detection, the trajectory uncertainty and the effect of a collision in the orientation of the robot are simulated in this work (Figure 3).

Figure 2. Image of Robonet.

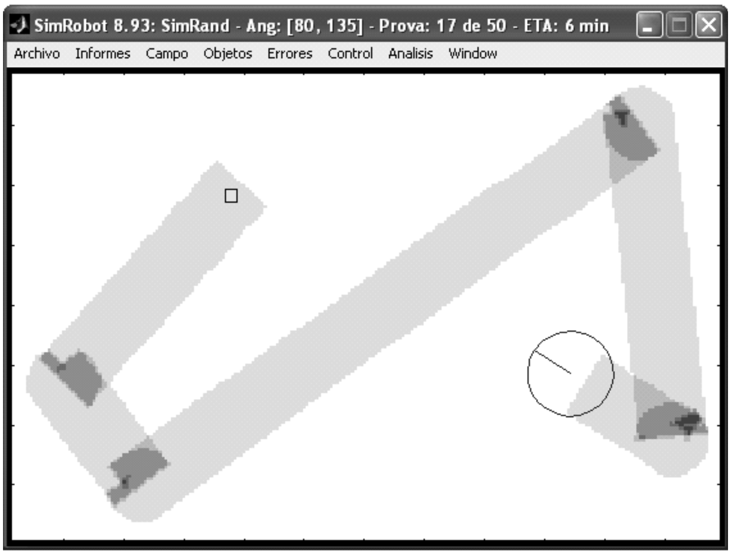

Figure 3. Simulation of Robonet.

4 Optimal Path-Planning Algorithm

Figure 1 proposes a basic random path-planning algorithm: move forward until collision then move back and turn a random angle. The probability density function of the turn angle is the only parameter that can be optimally adjusted. To this end, a first set of simulations have been defined over a rectangular scenario of 240x160 cm without obstacles and with a control algorithm based on Figure 1. Each cleaning operation is 20 minutes long and is repeated 10 times with a random starting position.

Figure 4 shows the number of collisions depending on the fixed turn angle. When the robot starts a movement the power consumption of the drive motors has a maximum peak so reducing the number of collisions will improve battery operation. To minimize the number of collisions the turn angle must be greater tan 80º. Figure 5 shows the largest forward distance run depending on the fixed turn angle. This distance can be used in some cases to have an estimation of the scenario size. To assure a very large forward cleaning the turn angle must be between 75º and 150º. Finally, Figure 6 shows the cleaned area depending on the fixed turn angle for two sample times: 1 and 5 minutes. To maximize the cleaned area after 5 minutes the turn angle must be greater tan 45º and lower tan 135º.

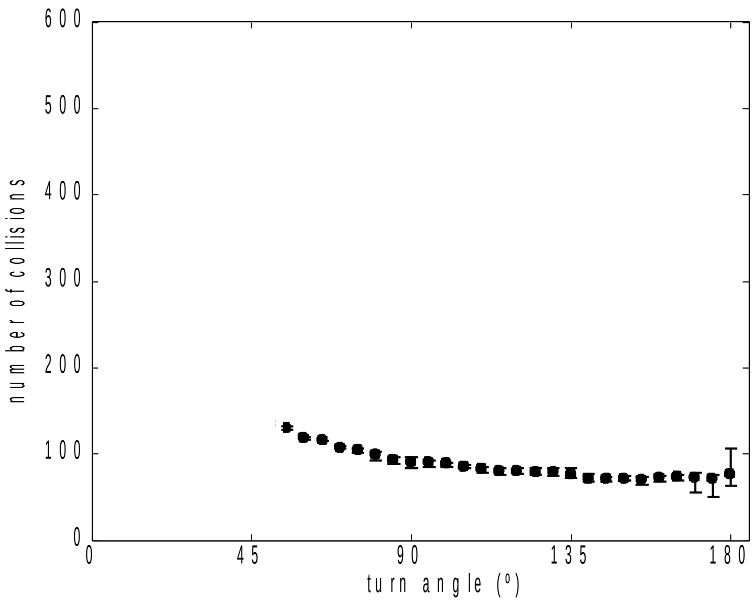

Figure 4. Maximum, minimum and average (dot) number of collisions depending on the fixed turn angle.

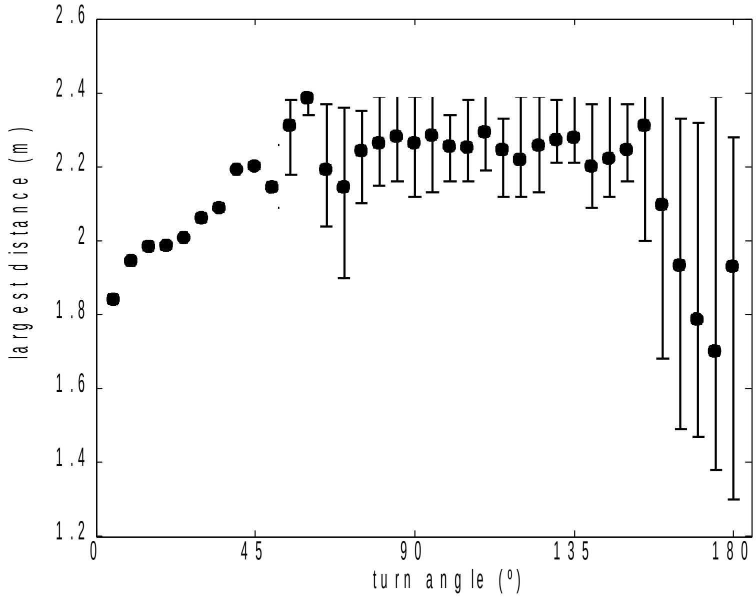

Figure 5. Maximum, minimum and average (dot) largest forward distance depending on the fixed turn angle.

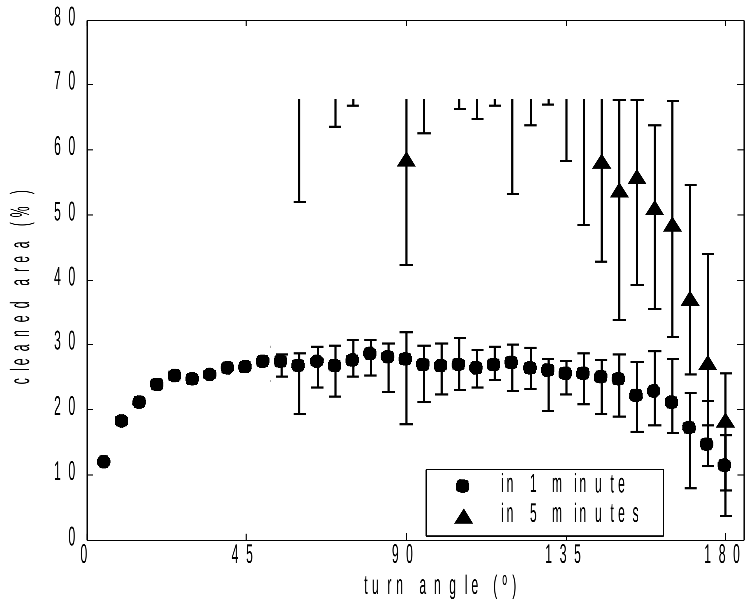

Figure 6. Maximum, minimum and average cleaned area obtained depending on the fixed turn angle and the time.

Therefore, the turn angle limitations obtained with the previous simulations can be used to define an optimal random path-planning algorithm just limiting the random turn-angle between 80º and 135º with an equally distributed probability density function. A second set of simulations have been defined to compare the proposed optimal method using the following case-control algorithms:

- **CASE A**: Random path-planning with turn angle between 0º and 180º.
- **CASE B**: Random path-planning with a turn angle from -180º to 180º.
- **CASE C**: Random path-planning with turn angle between 0º and 180º but using a turn angle opposite to the collision: if the collision is in the frontal-left part of the robot the turn is to the right and vice versa.
- **CASE D**: Proposed optimal random path-planning with a turn angle between 80º and 135º.
- **CASE E**: Proposed optimal random path-planning with turn angle between 80º and 135º but using a turn angle opposite to the collision: if the collision is in the frontal-left part of the robot the turn is to the right and vice versa.

Cases A, B and C are heuristic and must be considered as a reference. The proposed optimal random pat-planning algorithm is implemented in cases D and E. Cases C and E requires additional sensor information dealing with the relative position of the collision in the robot.

Figures 7, 8, 9 show the maximum, minimum and average results obtained when repeating 10 times a cleaning operation during 20 minutes with the selected control algorithms.

Figure 7 show the number of collisions depending on the case control algorithm. The case B is the worst so; it is not a good idea to have no limitations in the random turn angle. Additionally, there are small differences between the cases A and C so; this is not justified to include additional collision sensors only to convert the case A in the C. Finally, the minimum number of collisions is obtained with the cases D and E, based on our proposed optimal algorithm.

Figure 8 show the largest forward distance run by the robot in each experiment. The results are very similar regardless of the case. This information will be used lately to estimate the size of the cleaning scenario.

Finally, Figure 9 shows the evolution of the cleaned area. The results are very similar although the higher average and the less standard deviation in the 10 cleaning experiments correspond to case E algorithm. Therefore, considering the results shown in figures 7, 8, 9, we can conclude that the proposed random path-planning algorithm, implemented as case D or E, can be considered as optimum.

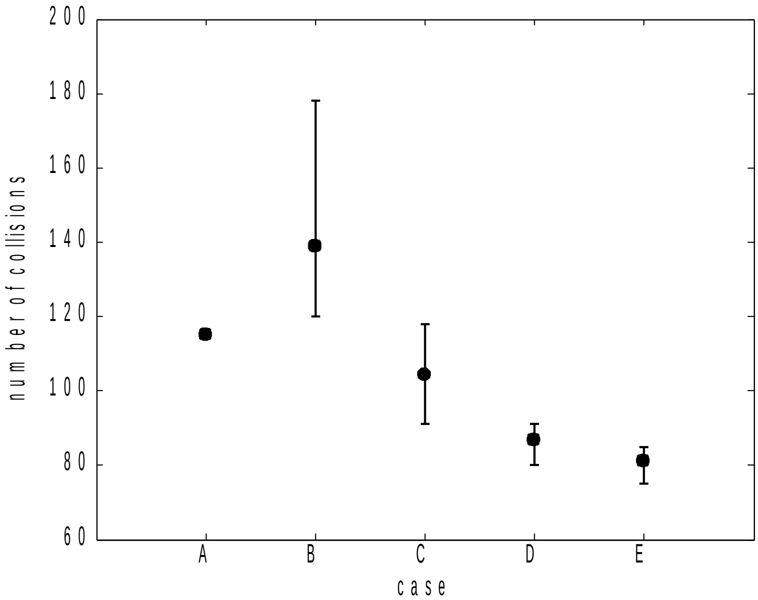

Figure 7. Number of collisions depending on the control algorithm.

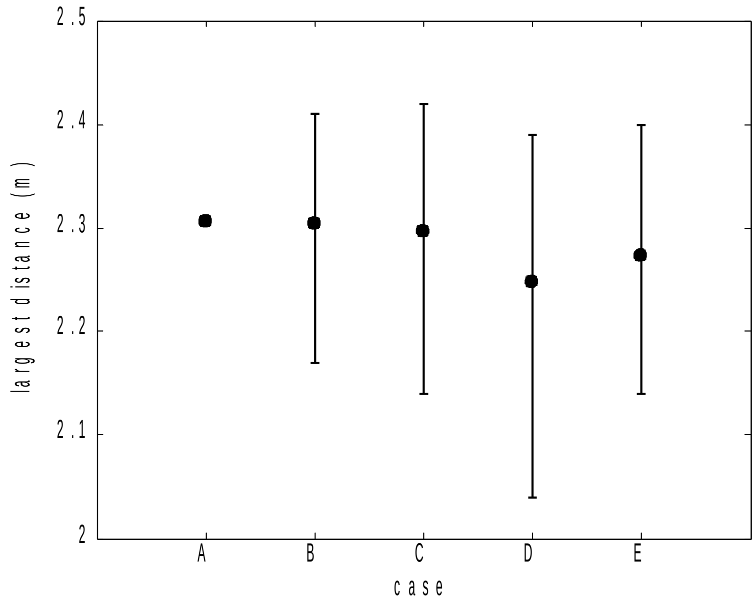

Figure 8. Largest forward distance depending on the control algorithm.

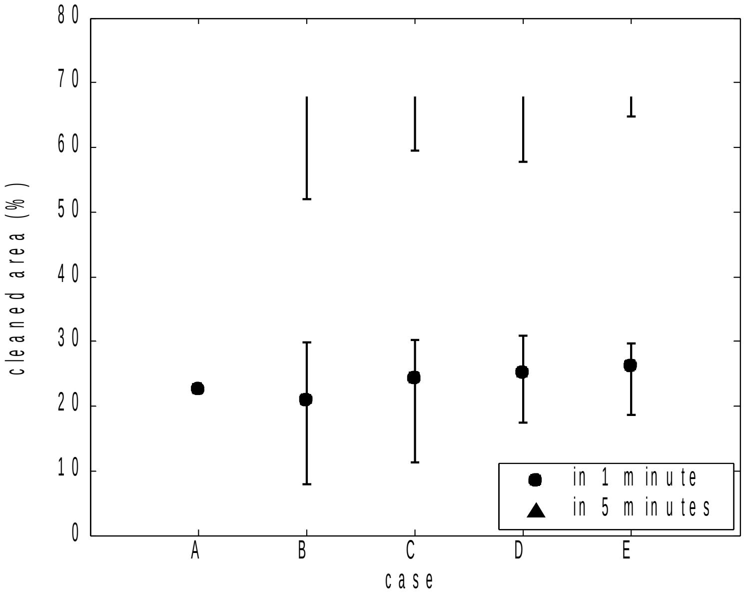

Figure 9. Cleaned area depending on the control algorithm.

5 Estimating the Size of the Cleaning Scenario

The size of the cleaning scenario is necessary in order to estimate the time that the robot must spend in the cleaning using the proposed optimal path-planning algorithm (case E).

5.1 Ideal Case

The ideal case corresponds to cleaning of an empty room, without objects or furniture. In the practice, there are very few specialized cleaning applications matching the ideal case constraints. Nevertheless, the ideal case is the first logical approach to the problem of estimating the size of the cleaning scenario.

To estimate the size without additional external sensors we propose to record the robot movements; mainly the turn angle after a collision and the forward distance until next collision. Then, as a first approach, the largest forward distance can be identified as the diagonal of a square scenario. Additionally, Figure 10 shows the histogram of all the forward distances ran by the robot in one cleaning experiment; with this information the largest distance and the most frequent distance (circles in Figure 10) can be identified as the length and the width of a rectangular scenario.

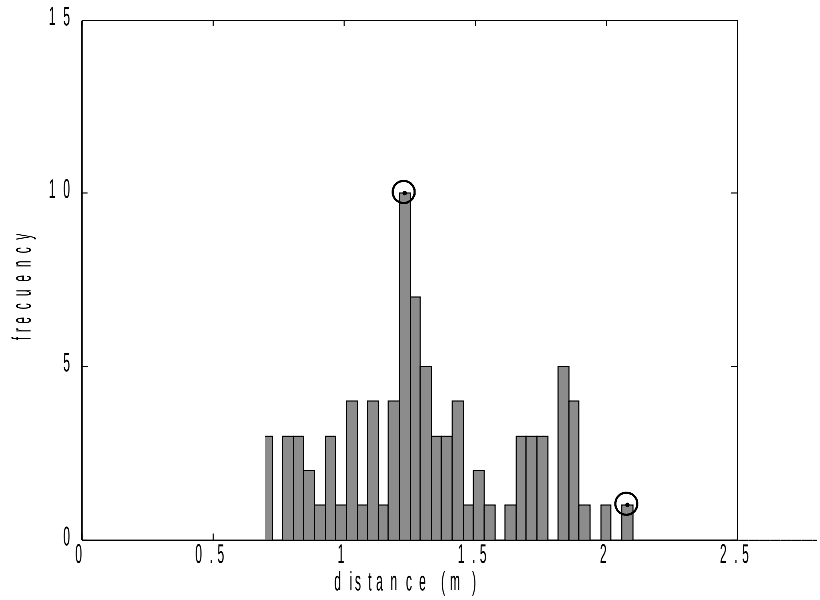

Figure 10. Forward distances run by the mobile robot in one cleaning. The most frequent distance and the largest are labeled.

Finally, as a third approach, the trajectory followed by the robot can be used to estimate trigonometrically the size of the cleaning scenario. To this end, the mobile robot must have left and right frontal collision sensors. Figure 11 shows an application example of the proposed procedure:

i) The method starts after a right-frontal collision (A in Figure 11), if the next collision is also from the right we can assume that the robot is in a corner.

ii) If the following collision occurs in alternative sides of the front part of the robot: left, then right, next left and so on; we can assume that the robot is between two parallel sides of the scenario.

iii) This sequence ends with two collisions on the same side of the front part of the robot (B in Figure 11). Despite the odometry errors, the angles and distances between collisions A and B can be used to compute DL and DT; the length and the width of the scenario.

Figure 12 shows the histogram of these two distances obtained in one cleaning operation without obstacles.

Figure 13 shows the average relative error obtained when estimating the area of four different clean rectangular scenarios using the three proposed methods. Each scenario cleaning is repeated 50 times. The minimum error (and even standard deviation) is reached when the size is obtained by trigonometric analysis of the robot trajectory. Remember that collision robot misalignment and trajectory errors are also simulated; otherwise the error is below 0.5%.

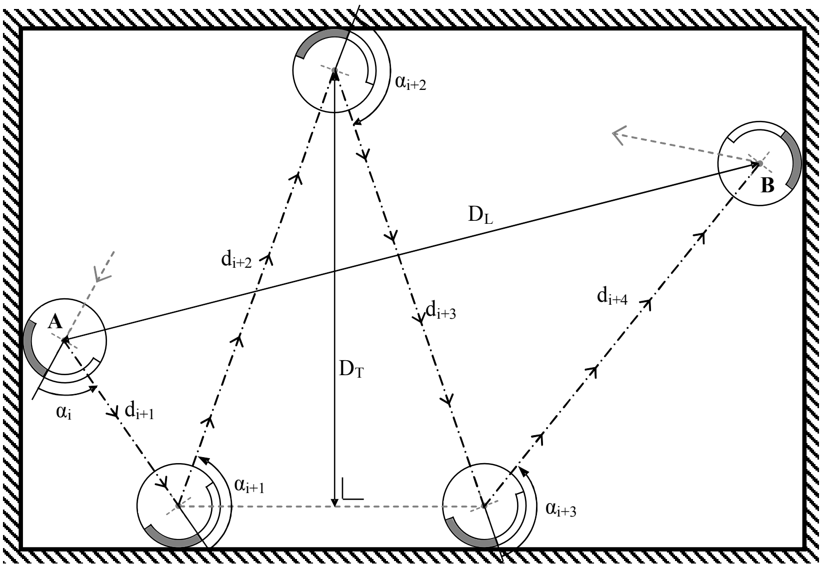

Figure 11. Special trajectory for size estimation. The collision (left or right side of the robot) is in gray.

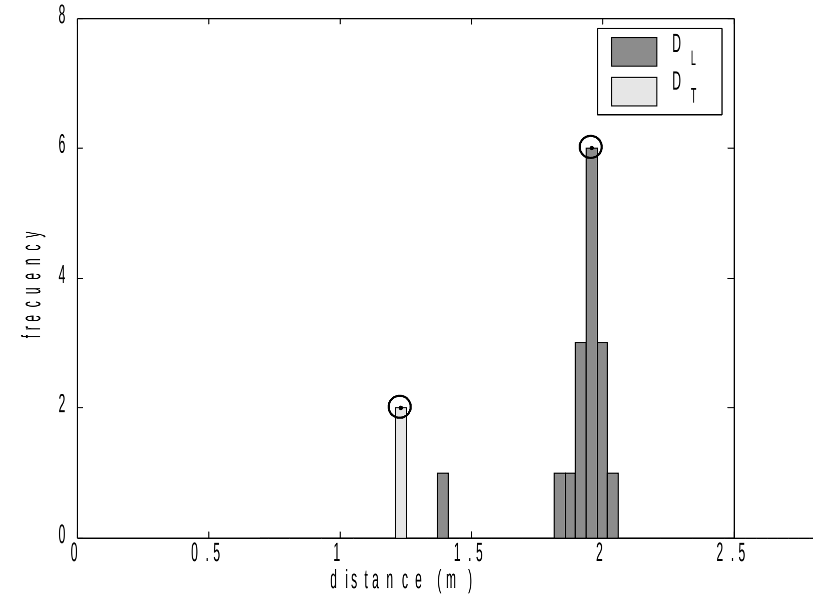

Figure 12. Histogram of DL (gray) and DT (white) distances obtained in one cleaning procedure. The most frequent values of booth values are labeled.

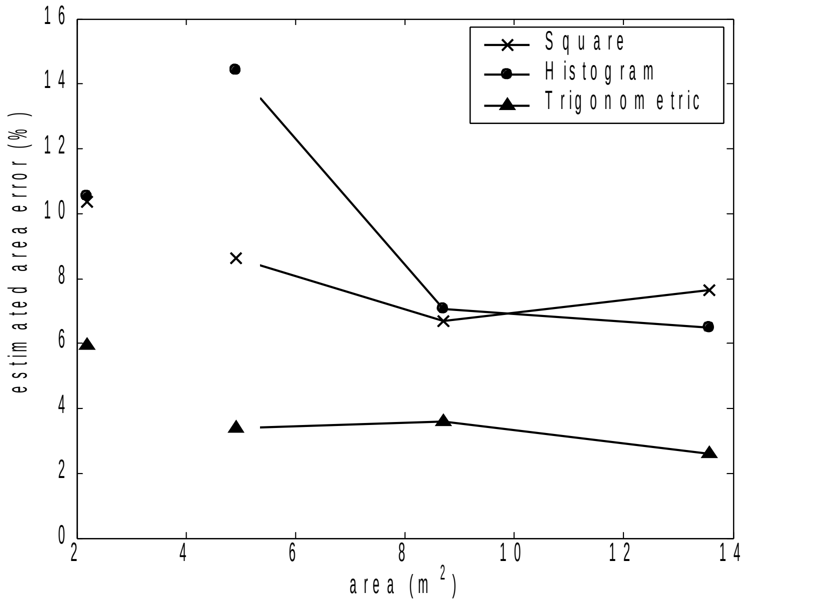

Figure 13. Average relative estimated area error depending on the size of the scenario and the estimation method.

The proposed methods take advantage of the fact that all collisions locates the walls of the room but, in a real case with objects or furniture distributed in the room, this supposition is not true.

5.2 General case

In the general case, real rooms can have objects or furniture. The experience obtained in the ideal case is used to propose a set of algorithms compatible with a basic random path-planning cleaning algorithm. In a normal room, it is supposed that most of the furniture and/or decoration are close to the walls with a clean (or passing) area in the center of the room and all the proposed algorithms will try to take advantage of this constrain.

5.2.1 Proposed Algorithms

5.2.1.1 Diagonal Maximum (DM)

The diagonal maximum algorithm is the same as described in the previous section, storing the maximum forward distance from collision to collision. The interpretation of this algorithm assumes that the shape of the room is squared and the maximum forward distance followed by the robot is the diagonal of the square area resulting in an underestimation of the area in case of non-square room. This algorithm is proposed again as a baseline algorithm.

5.2.1.2 Cross Exploration (CE)

The cross exploration requires to interrupt the main algorithm and execute the path of a simple cross (Figure 14). The uncertainty accumulated making the cross can preclude the recuperation of the original position of the robot and then this algorithm is only compatible with a main random path planning algorithm. The path of the cross exploration is as follows (Figure 14):

i) After a collision (point A) the robot starts the cross and turns a random angle, α_i, and goes forward until collision (point B).

ii) The robot turns 180º and goes back to the center of the previous distance (point C).

iii) The robot turns 90º to the left and goes forward until collision (point D).

iv) The robot turns 180º and goes forward until collision (point E) ending the cross exploration. This path produces a set of positions and distances d_{AB}, d_{BC}, d_{CD} and d_{DE} affected by the uncertainty in the trajectory of the robot however, at this stage, this uncertainty is not considered.

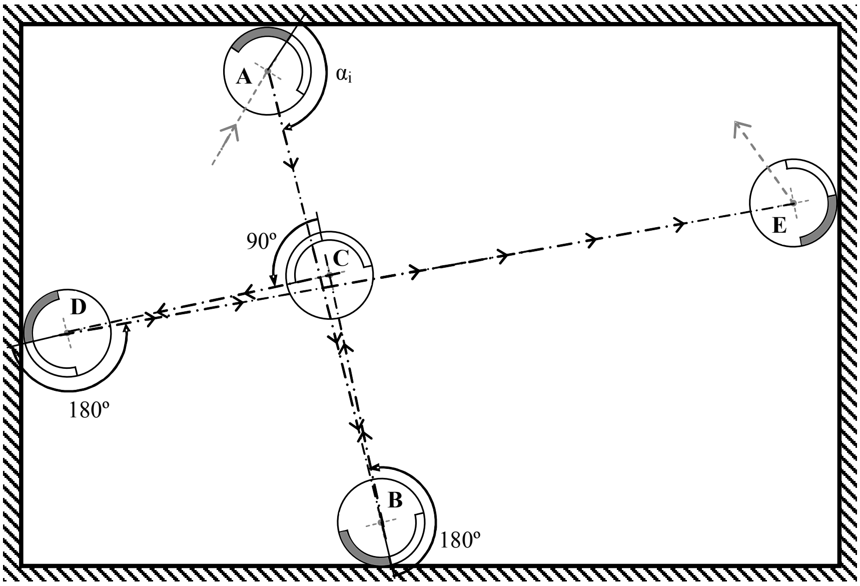

Figure 14. Example of "cross exploration". The cross starts in A, following the points B, C, D and ending in point E.

Unfortunately, in an empty room, the area computed as $d_{AB} \bullet d_{DE}$ can be up to 100% higher than the real area if d_{DE} coincides with the diagonal of the real rectangle that defines the room. To overcome this problem, a set of extreme rectangular areas that can be fitted outside the contact points A, B, D and E are defined (Figure 15). In each case, the area is computed and the average is used as the estimation of the area of the room. However, the cross exploration algorithm must be repeated a number of times to obtain an acceptable estimation of the area.

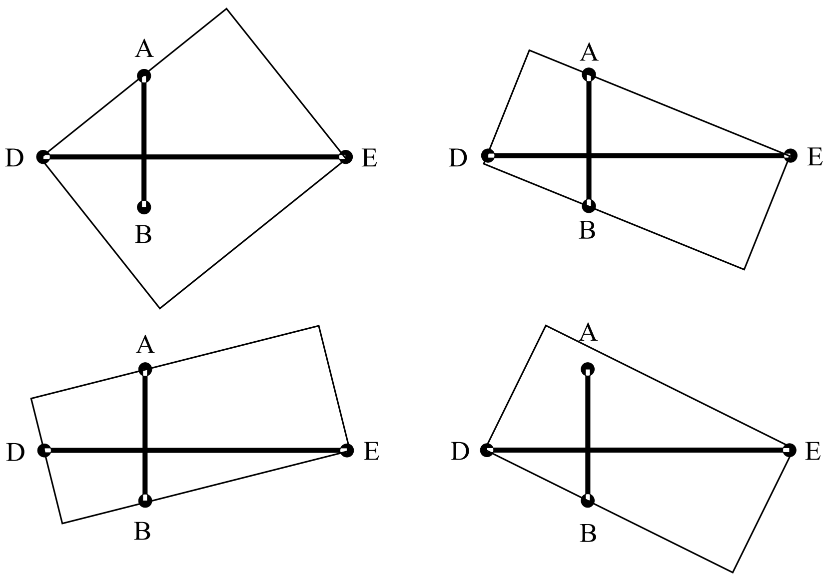

Figure 15. Example of rectangular areas that can be fitted outside the contact points A, B, D and E.

5.2.1.3 Square Exploration (SE)

The square exploration algorithm is performed at the beginning of the cleaning procedure and is based on the exploration of the perimeter of the room. The path of the square exploration is based on a sequence of inverted U as follows (Figure 16):

i) At the start the robot goes forward until collision (Figure 16-A).

ii) The robot turns 180º, goes back a fixed distance d1, turns 90º to the right, goes forward a fixed distance a_1, and turns 90º to the right.

iii) The robot goes forward until collision (measuring the distance b_2) and turns 180º.

iv) The distances d_1, a_1 and b_2 are used to compute d_2 and a correction angle β_{C1}.

v) The robot goes forward a distance d_2 and turns 90º-β_{C1} to the right (Figure 16-D) to stay parallel to the wall at a distance d_1.

This inverted U is repeated following the perimeter of the room and correcting the position of the robot when it seems parallel to a wall. The relative position of the collisions is stored for later use. This exploration ends when the trajectory of the robot goes near the starting point or at a given exploration time.

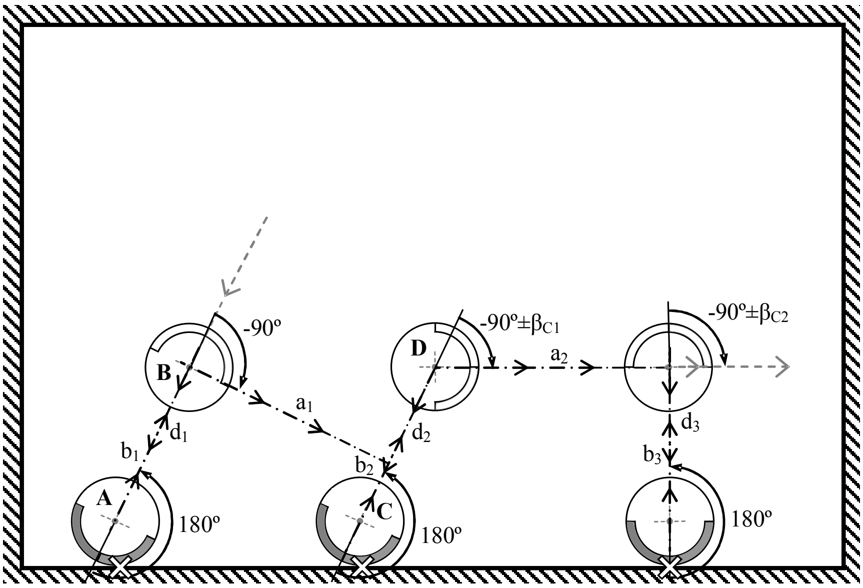

Figure 16. Example of "square exploration": A is the starting point.

After the exploration, the positions of the collisions are rotated (Figure 17) and the most external vertical and horizontal points are used to estimate the width and height of the scenario (similar to the Hough transform [26]). The minimum area obtained in all the rotations is selected as the estimated area.

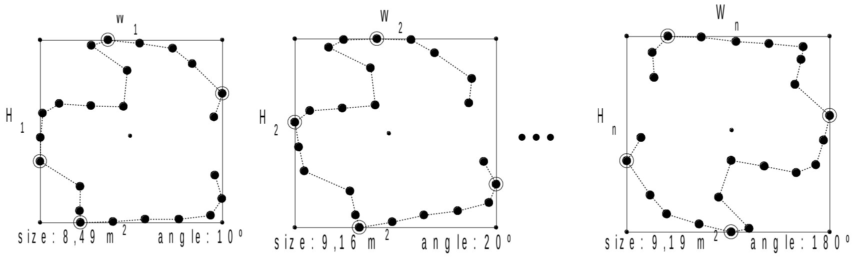

Figure 17. Example of the rotation process used to obtain the minimum rectangular area that fits the collision points (bedroom with objects).

5.2.1.4 Triangular Exploration (TE)

The triangular exploration algorithm is very similar to the square exploration but, in this case, the main exploratory movement is based on a triangular evolution as follows (Figure 18):

i) At the start the robot goes forward until collision (Figure 18-A).

ii) The robot turns 90º to the left, goes forward a fixed distance d_1, and turns 90º to the right.

iii) The robot goes forward until collision (measuring the distance b_1).

iv) The distances d_1, and b_1 are used to compute the correction angle β_{C1} and d_2.

v) The robot turns 90º+β_{C1} to the left and goes forward a distance d_2 (Figure 18-D). This triangle is repeated following the perimeter of the room and correcting the position of the robot when it seems parallel to a wall. The relative position of the collisions is stored for later use as in the previous algorithm.

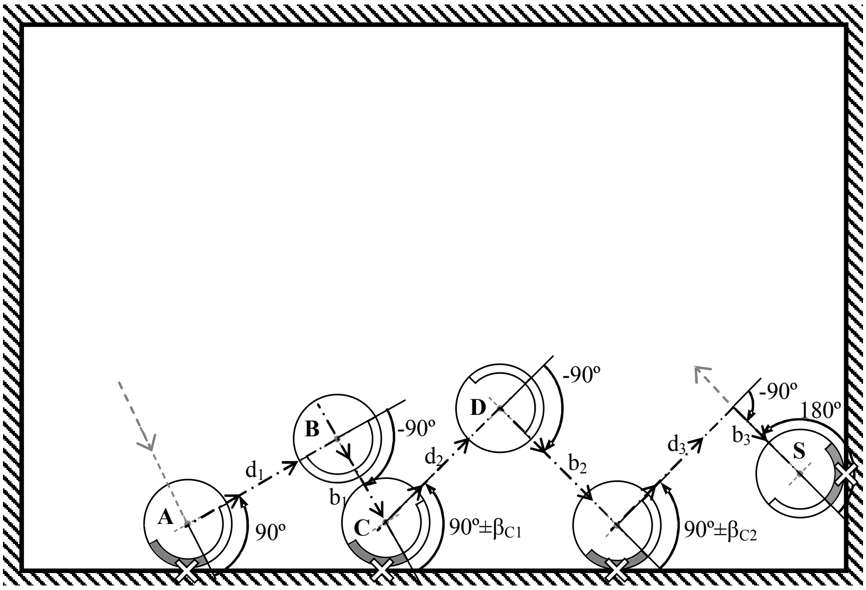

Figure 18. Example of “triangular exploration”: A is the starting point.

5.2.2 Simulation Results and Discussion

A set of simulations and scenarios (Figure 19) were selected to test the proposed methods for room area estimation: an individual office, a bedroom, and a dining room. Figure 8 shows the distribution of the furniture at a ground level; the areas in black and dark gray are inaccessible for the cleaning, the areas in soft gray are under objects but they are accessible for the robot. Table 1 shows the area of the selected scenarios, all of them are exact copies of real rooms in terms of size and furniture distribution and other objects as beds, sofas, chairs, unordered shoes, etc. Additionally, the simulations include trajectory uncertainty parameters measured in the real robot.

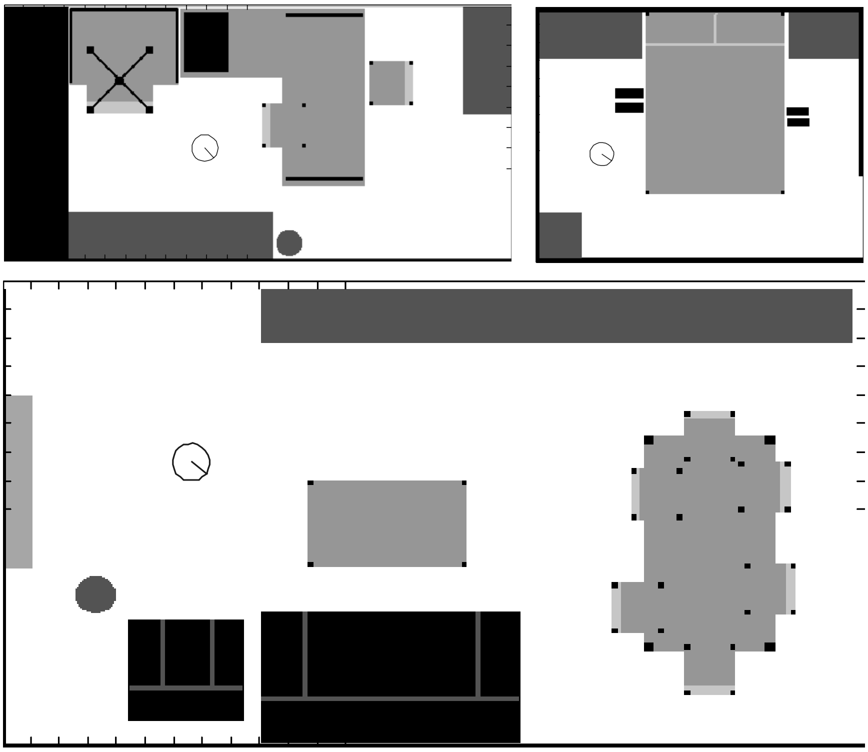

Figure 19. Cleaning scenarios: office (upper left), bedroom (upper right), and dining room (lower). The size of the robot is labeled with an empty circle with a line.

	Individual office	Bedroom	Dining room
Empty	10.58 m^2	9.42 m^2	19.13 m^2
With objects	8.66 m^2	8.12 m^2	15.10 m^2

Table 1. Cleaning area of the selected scenarios.

The scenarios selected are non-square and then the estimation error for the CE case without objects can be deduced analytically: +17.2% for the office, +3.4% for the bedroom (is almost square) and +20% for the dining room. But, these values are achieved if the robot goes exactly from one extreme of the diagonal to the other; very difficult if the exploration algorithm is based on a random path-planning.

Figure 20 and Figure 21 show the estimation error obtained with the proposed methods repeating each case 50 times and starting in a random position. The most interesting result is the large value and non-repetitive behavior of the error although, as the cleaning is a repetitive task, the average values should be considered as indicative. When the room is empty (Figure 20) the worst dispersion of the error corresponds to TE in the dining room; this is because of the uncertainty accumulated in the position of the robot that is bigger for large areas. The average error values are under ±10% for the CE, SE and TE algorithms. The SE algorithm (very similar to TE) has fewer problems because its square trajectory is more robust and easy to recover the alignment of the robot when it moves in parallel to a wall. However, the furniture or objects in the walls precludes any kind of alignment and the dispersion of the error increases (Figure 21). In the case of the office, plenty of unordered objects, all the proposed methods underestimate the area. However, if the area is the same but the central part of the room is empty (bedroom case) the error is reduced, especially for the CE algorithm. When the room have objects the area is underestimated in all the proposed methods; the CE has a slightly better average estimation error from -10% to -50%.

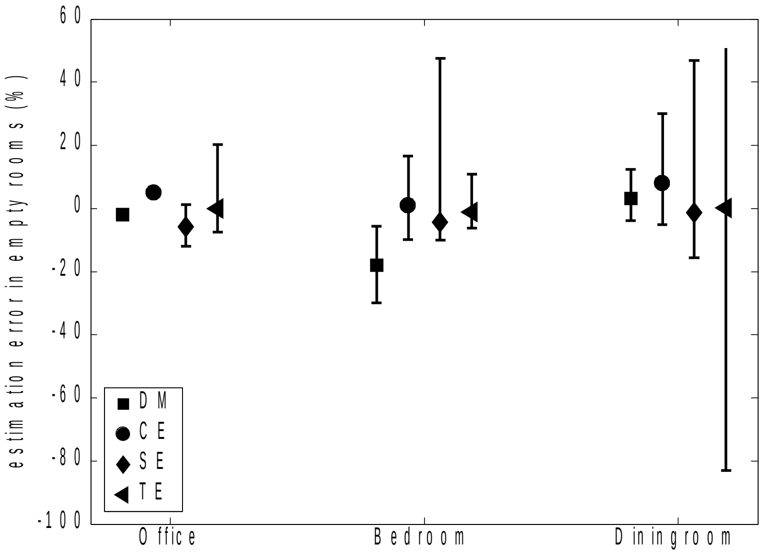

Figure 20. Maximum, minimum and average error in the estimation of the area of an empty room depending on the algorithm: DM, CE, SE and TE.

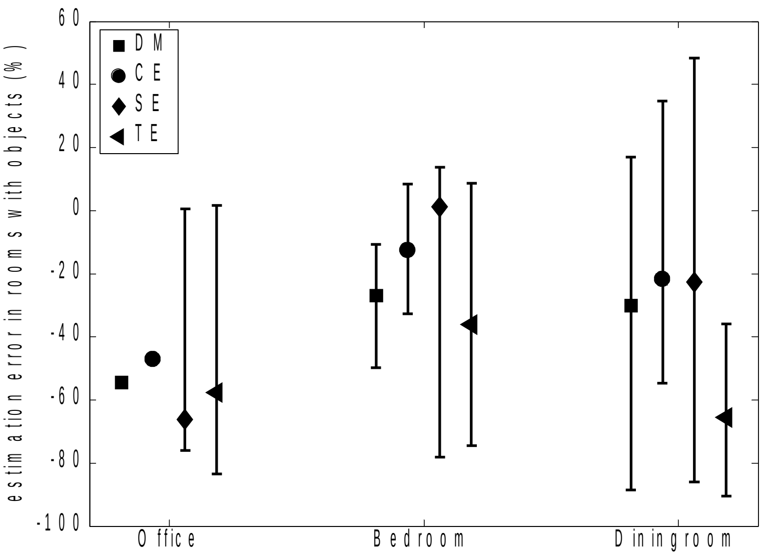

Figure 21. Maximum, minimum and average error in the estimation of the area of a room with objects depending on the algorithm: DM, CE, SE and TE.

Figure 22 shows the percentage of exploration cases where the algorithm SE and TE was blocked entering an infinite loop and the exploration of the perimeter cancelled. In all cases TE is highly sensitive to the furniture, for example entering easily under a chair, whereas SE is more robust because always tries to surround the objects.

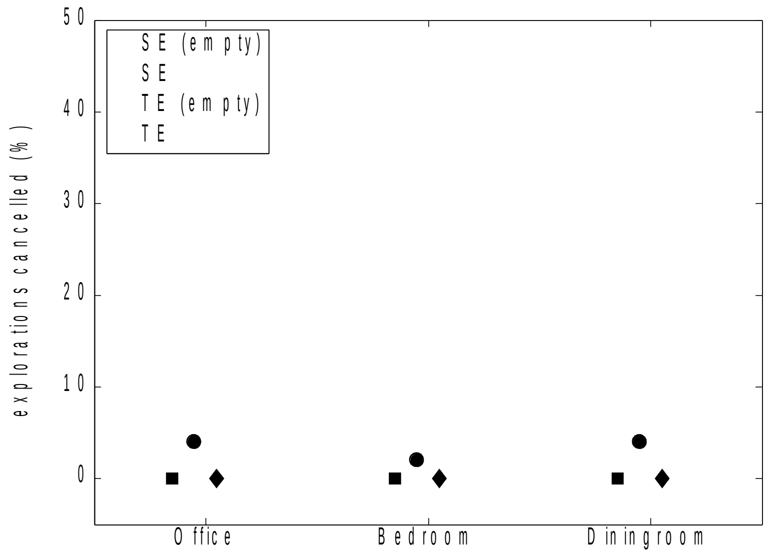

Figure 22. Percentage of explorations cancelled for the SE and TE methods in the cases without and with furniture.

Additionally, Figure 23 shows the time spend in the exploration process. Some of them are very small because the algorithm is confused, following a very short perimeter (Figure 24); in this case the estimation error can be up to -90%. Indeed, some exploration times are very large because the algorithm was confused by the uncertainties and the robot hardly reaches the starting point after several turns to the perimeter; in this case the estimation error can be up to +50%.

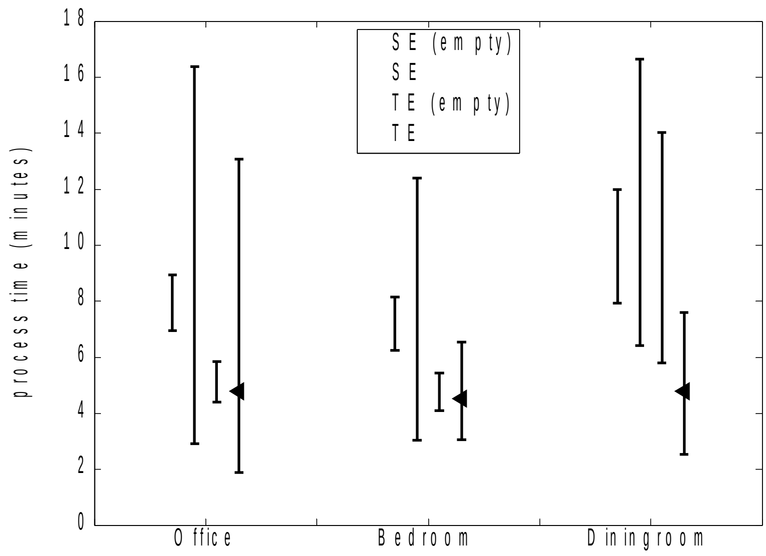

Figure 23. Time spend in the SE and TE methods.

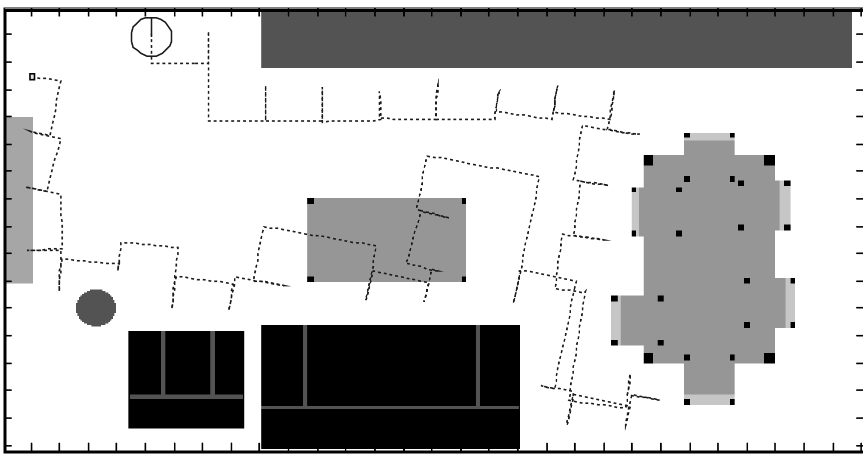

Figure 24. Example of real exploration trajectory for the SE method.

Finally, Figure 25 shows the evolution of a typical example of area estimation with the CE method. The worst drawback of CE is that at least 20 iterations were needed to obtain a representative estimation of the area, making this algorithm time consuming.

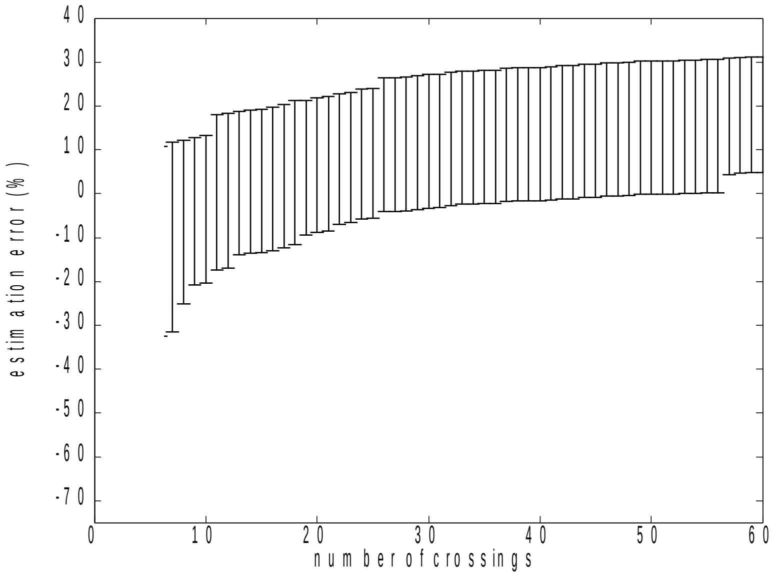

Figure 25. Evolution of the estimation error depending on the number of crossings: maximum, minimum and average (dot) values after 50 simulations.

6 Estimating the Evolution of the Cleaned Area

The last question to be answered is when the robot must be stopped. Some commercial cleaning mobile robots stop at a fixed amount of time or when the batteries are low. However, Figure 26 shows the evolution of the area cleaned at least one time by the robot. This evolution can be modeled with one exponential using a least squares method to fit (Figure 26 black line):

$$cleaned\_area(t) = A_{MAX} \cdot \left[1 - e^{-Td/t} \right], \qquad (1)$$

A_{MAX}: Maximum area (in m^2) cleaned by the robot

Td: Time constant (in seconds) of the evolution.

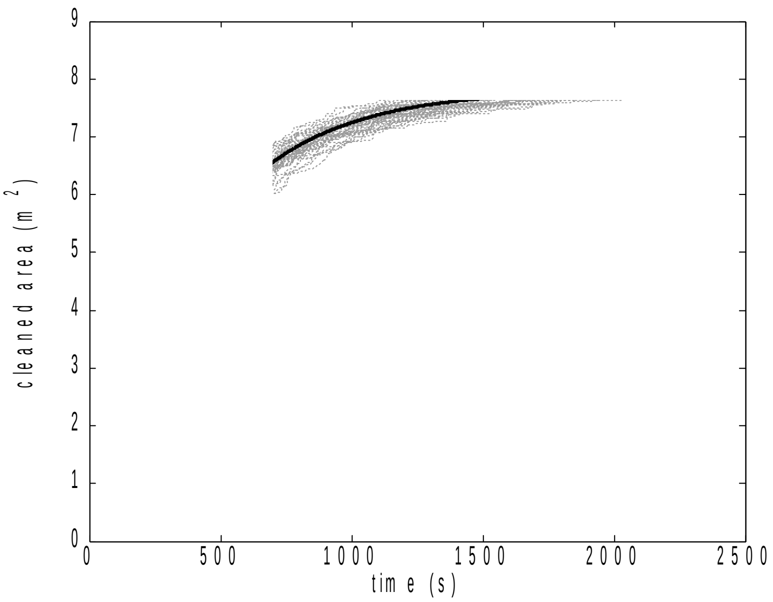

Figure 26. Evolution of the area cleaned by the mobile robot: 20 operations (gray) and adjusted by LMS (black).

Figure 27 shows the relative value of A_{MAX} depending on the total area of an ideal empty scenario. This value must be considered as an average value and it is interesting to notice that the cleaning operation never will reach the 100% of the scenario size. This is because of the particular structure of these cylindrical cleaning robots where the cleaning device is always shorter than the diameter of the robot, leaving a small area under the robot uncovered.

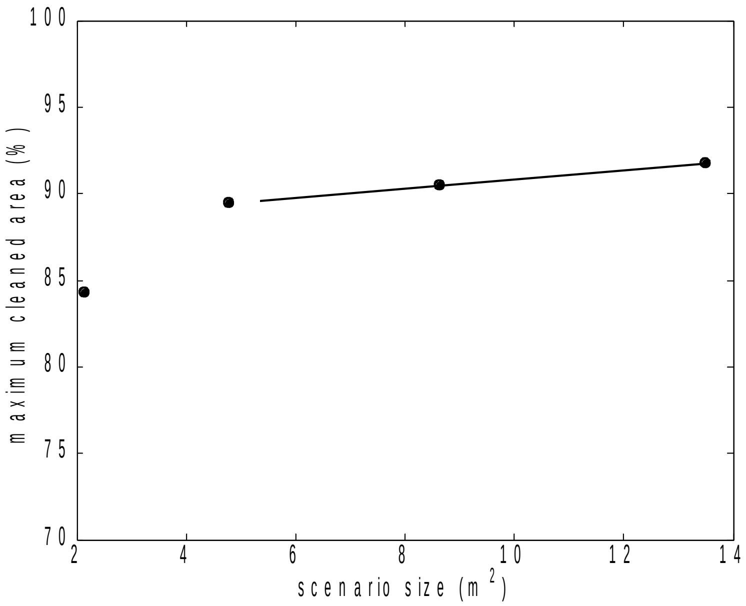

Figure 27. Relative value of A_{MAX} depending on the scenario size.

Figure 28 shows the value of Td depending on the scenario area (in m^2). These results indicate a linear relation between this parameter and the scenario size:

$$Td = 41.863 \cdot scenario\_size + 19.763 , \qquad (2)$$

Therefore, for the simulated robot, if the scenario size is known a priory or estimated during operation; the maximum area cleaned and the evolution of the cleaning can be estimated without additional sensors. The value of Td itself gives additional information about the time needed for complete coverage because it is known that the exponential evolution reach the 99% of its maximum value at 5•Td. However, the main drawback is that a large number of simulations or experimental measurements are needed to obtain A_{MAX} and Td for a given mobile robot.

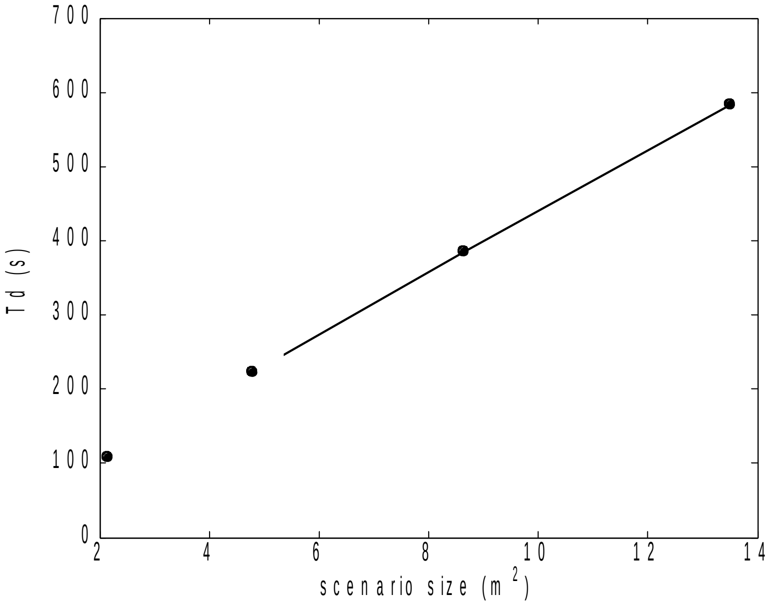

Figure 28. Relative value of Td depending on the scenario size.

7 Conclusions

In this work a method to estimate the coverage performances of a floor cleaning random path-planning mobile robot is introduced. A mobile robot simulator developed in MATLAB was used in the analysis and the main conclusions obtained are:

An optimal random path-planning algorithm based on a forward movement until collision and a random turn can be defined just limiting the turn-angle between 80º and 135º.

A method to obtain the size of an ideal empty scenario based on the trigonometric analysis of a specially defined trajectory of the robot has been proposed; obtaining an error less than 6%.

In a general case with furniture, the simplest method to estimate the room size is based on using the large forward distance ran by the robot as the diagonal of a square area. Average estimation errors under ±20% can be expected for empty rooms and from 0% to -70%. But this error depends on the path-planning algorithm of the robot.

The uncertainty in the robot position precludes the application of complex methods based on the analysis of the trajectory of the robot.

The maximum area cleaned by a cylindrical robot never reaches the 100% of the available area; the expected value is from 90 to 95% because the cleaning device is always shorter than the diameter of the robot, leaving a small area uncovered under the robot.

The cleaning evolution can be estimated using a single exponential model where the time constant can be expressed as a linear function of the scenario size. However, at this point, the parameters of this linear function are robot dependent and must be obtained through simulation or experimentation.

The time constant of the exponential model and the scenario size can be used to estimate the time needed for complete coverage without any other sensor or manual operation involved.

The estimation of the real room size obtained with a low cost cleaning mobile robot with only encoders and collision detectors is good in average so, if the robot uses this value to stop the cleaning process, the frequency of the cleaning must be higher than using manual methods to assure an average coverage of the cleaning.

8 Bibliography

[1] R. N. De Carvalho, H. A. Vidal, P. Vieira, M. I. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," in *Proc. of the IEEE Int. Symp. on Industrial Electronics*, vol. 2, pp. 677 – 682, Jul. *1997*.

[2] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition." *Autonomos Robots*, Vol. 9, pp. 247-253, 2000.

[3] M. Dlouhy, F. Brabec, P. Svestka, "A Genetic Approach to the Cleaning Path Planning Problem," in Proc. of *Euro-CG-2000 - 16th European Workshop on Computational Geometry*, Eilat, Israel, March 2000.

[4] G. Nagib, W. Gharieb, "Path Planning for a Mobile Robot Using Genetic Algorithms," in *Proc. of the Int. Conf. on Electrical Electronic and Computer Engineering*, Cairo, Egypt, pp. 185-189, Sept. 2004.

[5] S. X. Yang, C. Luo, "A Neural Network Approach to Complete Coverage Path Planning," *IEEE Trans. on Systems, Man, And Cybernetics—Part B: Cybernetics*, vol. 34, no. 1, pp. 718-724, February 2004.

[6] J. A. Moreno, M. Castro, "Heuristic Algorithm for Robot Path Planning Based on a Growing Elastic net," *Lecture Notes in Computer Science*, vol. 3808, pp. 447-454, Springer-Verlag, 2005.

[7] Y. Gabrielly, E. Rimon, "Spiral-STC: an on-line coverage algorithm of grid environments by a mobile robot", in Proc. *of the IEEE Int. Conf. on Robotics and Automation (ICRA'2002)*, Washington, DC, USA, 11-15 May 2002, vol. 1, pp. 954-960.

[8] E. González, P. Aristizábal, and M. Alarcón, "Backtracking Spiral Algorithm: A Mobile Robot Region Filling Strategy," in *Proc. of the 2002 Int. Symp. on Robotics and Automation*, pp. 261-266.

[9] E. González, M. Alarcón, P. Aristizábal, and C. Parra, "BSA: A Coverage Algorithm," in Proc. *of the IEEE Int. Conf. on Intelligent Robots and System (IROS'2003)*, Las Vegas, USA, 27-31 Oct. 2003, vol. 2, pp. 1679-1684.

[10] E. González, O. Álvarez, Y. Díaz, C. Parra, C. Bustacara, "BSA: A Complete Coverage Algorithm," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'2005)*, Barcelona, Spain, 18-22 April 2005, pp. 2040-2044.

[11] H. Choset, "Coverage for robotics - A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113-126, Oct. 2001.

[12] J. Seop Oh, Y. Ho Choi; J. Biae Park; Y. F. Zheng, "Navigation of cleaning robots using triangular-cell map for complete coverage," *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'2003)*, Taipei, Taiwan, 14-19 Sept. 2003, vol. 2, pp. 2006-2011.

[13] S. X. Yang, C. Luo; Q. H. M. Meng, "Area-covering operation of a cleaning robot in a dynamic environment with unforeseen obstacles," *Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, July 2003, vol. 2, pp. 1034-1039.

[14] E. Acar, H. Choset, "Sensor-based coverage of unknown environments: incremental construction of Morse decompositions," *International Journal of Robotics Research* , Vol. 21, pp. 345-366, 2002.

[15] E. Acar, H. Choset, "Robust sensor-based coverage of unstructured environments," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS'2001)*, Maui, Hawaii, 29-3 Oct./Nov. 2001, pp. 61-68.

[16] J. Borenstein, L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." *IEEE Trans. on Robotics and Automation*, vol. 12, no. 6, pp. 869-880, Dec. 1996.

[17] A. Martinelli, "The odometry error of a mobile robot with a synchronous drive system," *IEEE Trans. on Robotics and Automation*, vol. 18 , no. 3, pp. 399-405, June 2002.

[18] J.A. Castellanos, J. Neira, J. D. Tardós, "Multisensor Fusion for Simultaneous Localization and Map Building," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 908-914, Dec. 2001.

[19] C. Estrada, J. Neira, J. D. Tardós, "Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments," *IEEE Trans. on Robotics and Automation*, vol. 21, no. 4, pp. 588-596, Aug. 2005.

[20] A. Burguera, G. Oliver, J. D. Tardós, "Robust Scan Matching Localization using Ultrasonic Range Finders," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and System (IROS'2005)*, Edmonton, Alberta, Canada, 2-6 Aug. 2005, pp. 1367-1372.

[21] M. Betke, L. Gurvits, "Mobile robot localization using landmarks," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 2, pp. 251-263, April 1997.

[22] J. Palacin, A. Sanuy, X. Clua, G. Chapinal, S. Bota, M. Moreno and A. Herms, “Autonomous Mobile Mini-Robot with Embedded CMOS Vision System.” in *Proc. of the IEEE Industrial Electronics Society Conf.*, Nov. 2002, vol. 3, pp. 2439-2444.

[23] J. Palacin, X. Lasa, S. Marco, “Straight-line path following in cleaning robots using lateral ultrasonic sensors,” in *Proc. of the IEEE Instr. and Meas. Tec. Conf.*, May 2003, vol. 2, pp. 1484-1487.

[24] J. Palacín, T. Palleja, I. Valgañón, R. Pernia, J. Roca, “Measuring Coverage Performances of a Floor Cleaning Mobile Robot Using a Vision System,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA’2005)*, Barcelona, Spain, 18-22 April 2005, pp. 4236-4241.

[25] M. Tresanchez, T. Palleja, J. Palacín, “Measuring Coverage Performances of a Floor Cleaning Mobile Robot Using a Vision System,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, California, 29 Oct-2 Nov 2007.

[26] P.V.C. Hough “Method and Means for Recognizing Complex Patterns,” U.S. Patent 3069654, Dec. 1962.