Algorithms for Maps Construction and Localization in a Mobile Robot

Daniel ROJAS[1], Ginno MILLÁN[2], Fernando PASSOLD[3], Román OSORIO[4], Claudio CUBILLOS[1], Gastón LEFRANC[1]

[1] Pontificia Universidad Católica de Valparaíso, Av. Brasil 2950, Valparaíso, Chile,
glefranc@ucv.cl

[2] Universidad Católica del Norte, Larrondo 1281, Coquimbo, Chile,
gmillan@ucn.cl

[3] Universidade de Passo Fundo, BR 285, Bairro Passo Fundo, Sao José, Brazil,
fpassold@upf.br

[4] IIMAS, Universidad Autónoma de México, Av. Universidad 3000, México,
roman@unam.mx

Abstract: In this paper it is presented an integration of algorithms that permits maps construction and navigation of mobile robots. Simultaneous Localization and Mapping (SLAM) algorithm is used based on FastSLAM method. Navigation system is based on Vector Field Histogram algorithm to avoid obstacle and a spiral way trajectory method. Up to three different complex simulation maps have been used to evaluate the system.

Keywords: Mobile Robot, Vector Field Histogram (VHF) algorithm, Simultaneous Localization and Mapping (SLAM), FastSLAM.

1. Introduction

The main problem in an autonomous mobile robot is the complexity and difficulty to move in an unknown environment. To solve this problem, the robot must construct a map of the environment and, at the same time, determine their local localization on the map. If only a map is provided, it is called global localization. When only a priori state is given, both states the map and pose are estimated simultaneously, it is called simultaneous localization and mapping. This is the reason SLAM and global localization is considered disjointed problems. But, neither in practice are nor disjointed because of the map is imperfect, having small errors and a lack or an excess of landmark [1], [2].

Objects or obstacles that exist in an environment are called landmarks. In real applications, unknown environment exploration has a set of landmarks, allowing to measure relative distance from the robot to each landmark.

In this paper, both SLAM and global localization are used for mapping and a mobile robot simultaneous localization (SLAM). This permits a mobile robot moving in an unknown environment and its implementation in closed environments (e.g. rooms, offices, warehouses, etc.) with autonomous exploration [1], [2].

SLAM solutions have generally been based on Extended Kalman Filter (EKF) SLAM [3]-[5]; and type Rao-Blackwellized particle filter, called FastSLAM. [8].

EKF filter allows to integrate the measures carried out by the robot and control actions to create the probably environment map. Figure 1 shows a set of landmarks in the environment, the robot's position and observations on the landmarks. [9], [10].

This SLAM EKF algorithm grows quadratic form with the number of landmarks on the map, making complicated to generate maps. It is also sensitive to erroneous data associated to observations with landmarks.

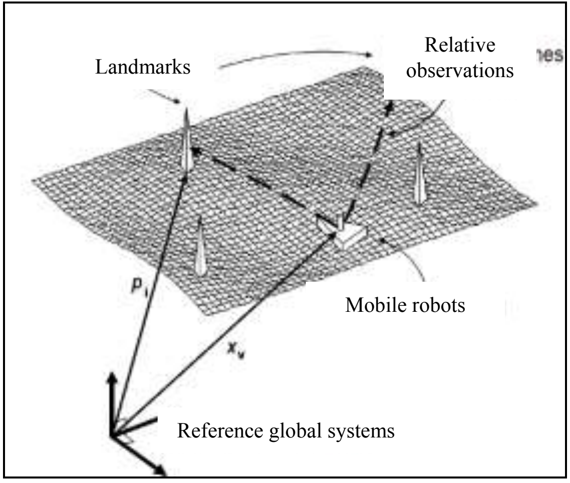

Figure 1. SLAM using the Kalman filter.

It has been proposed a compressed algorithm to improve performance and reduce complexity. Map is subdivided into several sub maps [4].

Online SLAM, recovers the present position X_t of the robot, instead of the entire route. These algorithms tend to be incremental and can process a data item at a time. This type of algorithms is usually called filters.

To solve any SLAM problem, the robot must have two models: a mathematical model that relates measurements of odometry u_t to X_{t-1} and X_t robot positions, and a model that relates measurements Z_t to the environment map and X_t robot location.

There are several methods for mobile robot navigation from one point to another one; it uses the VHF algorithm for navigation. This algorithm utilizes a two-dimensional Cartesian histogram to represent obstacles. Each cell has a value representing the certainty that there is an obstacle in that position. The Cartesian histogram reduces to a one-dimensional polar histogram. In this ways, the probability of finding an obstacle is obtained. Then, all available routes to the destination are look for, and select which have less cost function *G*. [11]-[14].

In this paper, an algorithm integration that allows map construction and simultaneous localization (SLAM), with autonomous indoor navigation through a mobile robot is presented. A construction maps and location algorithm is based on the FastSLAM method. For autonomous exploration, it uses the method of Vector Field Histogram (VHF) based navigation algorithm for navigation to avoid obstacles and a spiral shape path, to explore the environment. This algorithm integration is is evaluated using simulation in three different types of maps, with varying degrees of complexity. Then, developed algorithm is implemented in a real mobile robot.

2. SLAM Math Model

An autonomous mobile robot in unknown environment goes from a position with coordinates known, to reach a new position. Its movement is uncertain, making it difficult to determine their global coordinates. When it moves, the robot perceives their environment through sensors, and changes its movement to avoid obstacles, navigating until reaching the desired position. The SLAM problem is environment map construction and the determination of the relative position of the robot on the map. [2], [4], [5].

Formally, the SLAM is described in probabilistic terms. Robot location is X_t. and time is *t*. For mobile robots on a surface, X_t is usually a three-dimensional vector, composed of its two-dimensional coordinates in the plane, plus a turn unique value for orientation. The sequence of positions, or routes, is given as

$$X_T = \{x_0, x_1, x_2, \ldots, x_T\}, \tag{1}$$

where *T* is the final time (*T* may tend to infinity). The initial location x_0 is known. Other locations may not be detected. The odometry provides information between two consecutive points.

$$U_T = \{u_1, u_2, u_3, \ldots, u_T\}, \tag{2}$$

where U_t denotes the odometry characterized in the movement between the time $t - 1$ and *t*; such data are obtained from the wheel encoders or the control applied to the motors.

This sequence characterizes the relative motion of the robot. For noise-free movement, the sequence U_T would be enough to retrieve the coordinates X_T since the initial location x_0. However, odometry measurements contain noise, and the route determined by the techniques of integration, it has inevitably an error.

For environment objects detection, by the robot, is *m* a true environment map. The environment can be composed of points of reference objects, surfaces, etc., and *m* describes the location the location of them. The environment map *m*, typically is assumed unchanged in time.

The robot measurements set information between features in m and robot location x_t. If it is assumed that the robot takes a measurement at each point in time, the sequence of the measurements is:

$$Z_T = \{z_1, z_2, z_3, \ldots, z_T\}. \tag{3}$$

The main methods of SLAM are the Extended Kalman filter, based on landmarks and the particle filter (FastSLAM).

The Extended Kalman Filter SLAM algorithm obtains a world model *M* and positions sequence X_T from odometry and data measurement. Pose *p*, position and orientation of a robot, is determined from equation

$$p(y_t \mid z_t, U_T) = p(x_t, m \mid z_t, U_T). \tag{4}$$

The Figure 2 shows a sequence of locations, sensors measurements, and the relationship between these variables. This graphic model allows the understanding of relationships and dependencies on SLAM [2].

Full SLAM algorithm is based on particles, i.e. in the robot position, using the EKF filter to estimate the objects.

The algorithm determines the position throughout robot trajectory, by calculating post probability of the set X_T and the map m from the data available.

The variables on the right-hand bar of equation (5) are directly observable to the robot; while on the left-hand are the desire one. Algorithms for Full SLAM are processed in batches, i.e. all data at the same time.

$$p(x_t, m \mid z_t, U_T, c_t) = p(x_t \mid z_t, U_T, c_t) \cdot \prod_{n=1}^{N} p(m_n \mid x_t, z_t, U_T, c_t), \tag{5}$$

where x_t is the position of the robot; and y_t is (x_t, m); m is the map; z_t are data from sensors; and U_T are control orders.

Figure 2 illustrates the graphical model of Full SLAM, where is obtained the path travelled by the mobile robot based on the set of positions X_T in the environment map m. [7], [13].

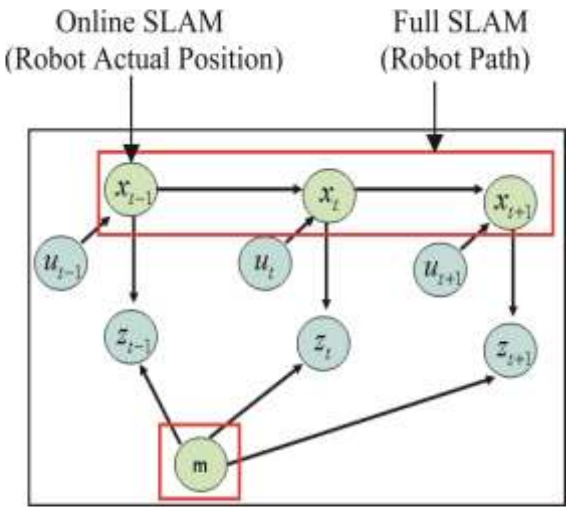

Figure 2. SLAM and Full SLAM graphic model.

3. SLAM System

In the map construction, the algorithm converts robot coordinates and the objects detected to a global system of coordinates (Figure 3). It is based on previous work [15].

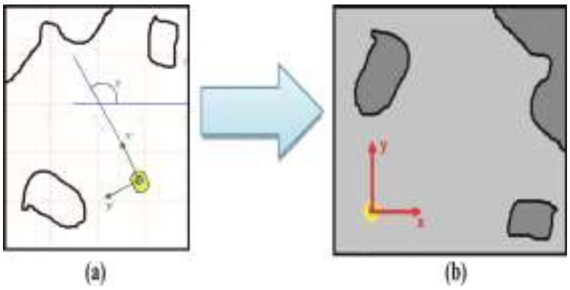

Figure 3. Global coordinate system.

It is used Vector Field Histogram algorithm for navigation Figure 4.

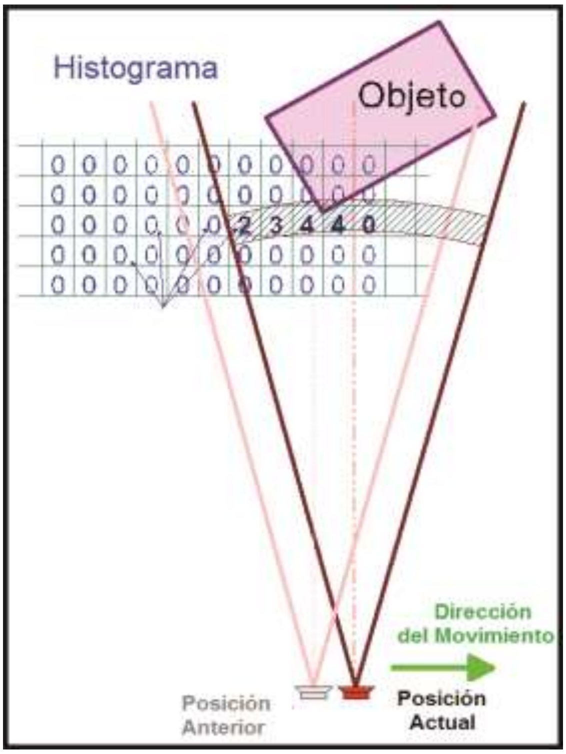

Figure 4. Navigation method.

Map is scanned in a spiral way using the VFH navigation to avoid obstacles (Figure 5).

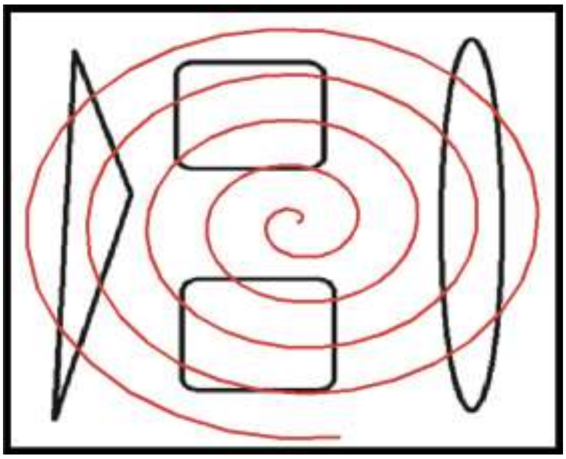

Figure 5. Robot exploration.

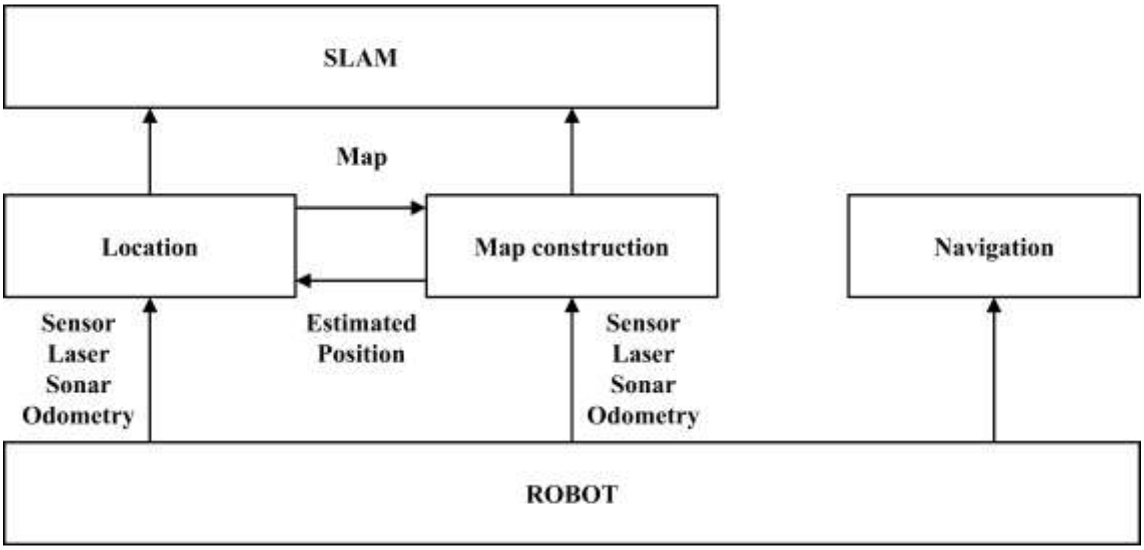

Figure 6. Integrated SLAM navigation system.

Figure 6 shows the integrated SLAM system with the navigation system, in a mobile robot.

The integrated system is simulated using the Player/Stage program as a tool. The Player is used as a server for robot control and can be programmed in C++, Java and Python. The Stage part allows to simulate aspects cinematic in two-dimensional and simulate data from the sensors (Figure 7). In this case, a Pioneer robot has been simulated.

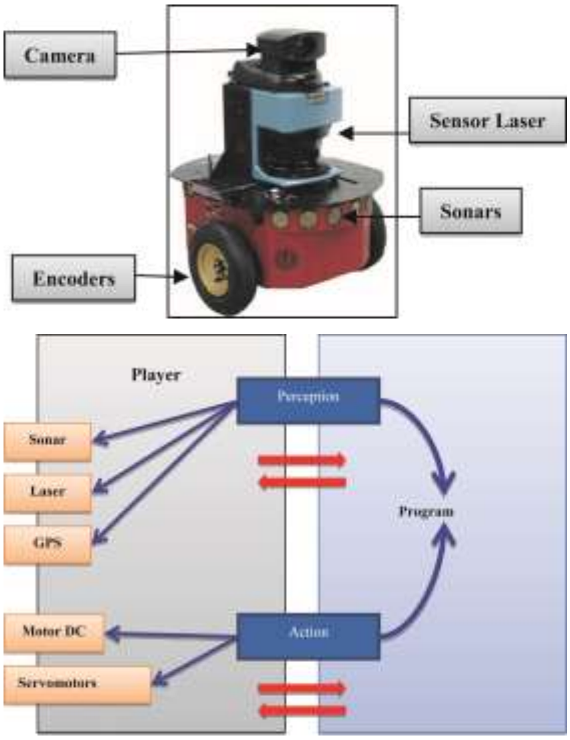

Figure 7. Pioneer robot simulated with Player/Stage.

Then, developed algorithm is implemented in a real mobile robot used in [16]. This robot has two wheels in differential configuration and it consists of 5 levels which contains the processing system and different sensors and actuators.

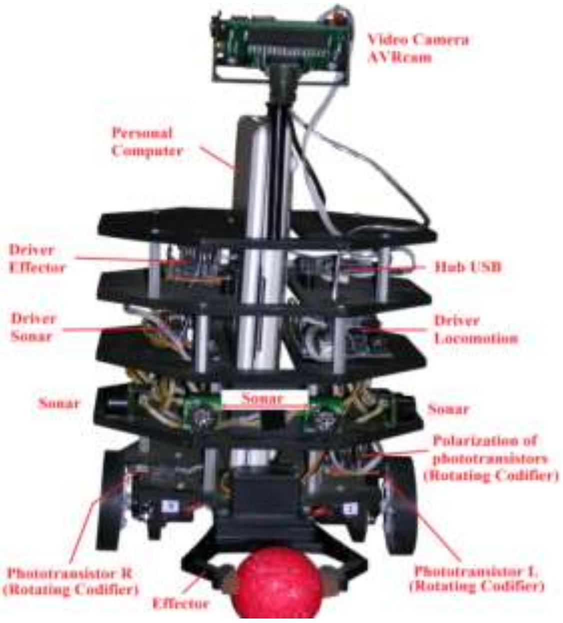

Figure 8. Real robot implemented.

4. System Evaluation

Evaluations are made in simulation environment using Player/Stage. Then, player stage is set in the real robot, using a small netbook connected to the sensors and actuators.

Integrated algorithms are evaluated on simulation, focusing primarily on the localization algorithm and then navigation algorithm.

In the case of location algorithm, parameters are adjusted defining the way in which uncertainty increases about the position of the robot making a movement in the movement model called "Sample Motion". It is possible to estimate these parameters, but, these values depend on a number of factors, such as: the type of soil of the environment, the wear of wheels, and the robot speed robot. In this case, parameters are adjusted in the simulation.

In VFH navigation algorithm, parameters are adjusted that allowing the robot to approach at a minimum distance of 10 cm to an object, to decreases the angular speed or rotation speed of the robot and reduce robot sudden movements that cause increased error in the estimation of the movement model.

In simulation, using Player/Stage, three scenarios are used to test: a map simple 1, 2 cave map, and a map 3, which is a maze. All scenarios have a 256 m^2 surface area. These maps can be seen in Figure 9.

The first map consists of a closed environment with simple obstacles with geometrical forms. The second map consists of a closed environment with obstacles more complex than the map 1, representing the caves. The third map is in a closed environment with obstacles, which represents a labyrinth.

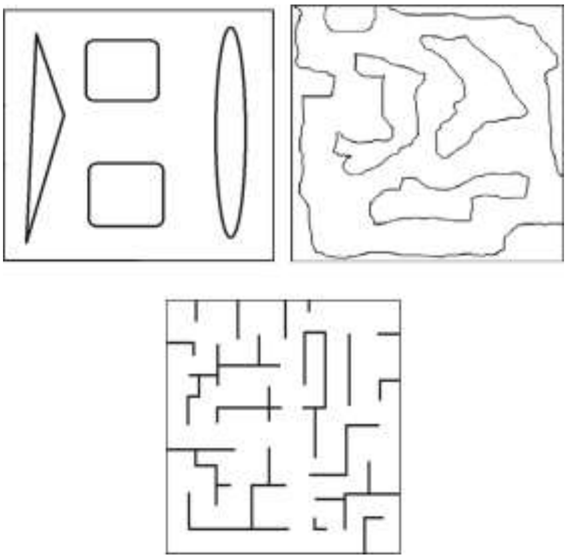

Figure 9. Maps 1, 2 and 3.

Exploration time in map 1 is approximately 12 minutes and a distance travelled is 120 [m].

It is observed that relative error reaches a maximum value of 12% and they appear only in certain particles. However, in most of the particles, the relative error of the estimated position of the robot is less than 2%. The estimate coordinate Y [m] has a greater error average with a value of 1.09%. Table 1 and 2.

Table 1. General data map 1

Map size	256 m^2
Trajectory time	12 minutes
Distance traveled	120 m

Table 2. Errors obtained with map 1

Estimated position	Relative error average (%)	Average absolute error
Position X m	0.78	2.39 cm
Position Y m	1.09	2.43 cm
Orientation rad	0.68	0.006 rad

The unit used to measure the robot position is metro, errors in each position are in the range among the 1 to 12 [cm] and robot size has a dimension of 40 [cm] wide by 40 [cm] long. Algorithms system worked satisfactorily in map 1.

Map 2 simulation obtains a trajectory time of 10 minutes and a distance of 113 [m]. The relative error peak is 12% error and appears only in certain particles. However, in most of the particles the relative error of the estimated position of the robot is less than 2%. The estimate in the coordinate is in [m], which has a greater error average value of 1.38%.

Map 2 results can be seen in tables 3 and 4.

Table 3. General data map 2

Map size	256 m^2
Trajectory time	10 minutes
Distance traveled	113 m

Table 4. Errors obtained with map 2

Estimated position	Relative error average (%)	Average absolute error
Position X m	0.81	1.26 cm
Position Y m	1.38	1.58 cm
Orientation rad	0.81	0.004 rad

Comparing data from the evaluation of map 2 and map 1, it is observed an increase in the number of peaks and the extent of the average relative error of particles on the map 2. But the average absolute error is slightly less than the map 1.

Considering that the magnitude of the errors at each position are in the range of between 1 [cm] to 12 [cm] and that the size of the robot has a dimension of 40 [cm] wide by 40 [cm] long, confirms that algorithms system worked satisfactorily in the map 2.

For X coordinate, relative error average is 0.34% and the absolute error of 0.05 cm, which is small. For Y coordinate, relative error is 0.53% and the absolute error is 0.07 cm. For orientation, the error is 0.43% and the absolute error is 0.001 radians.

In map 3 simulation, exploration time is approximately 30 minutes and distance is 245 [m]. The relative error peak is 12% and appears only in certain particles. However, in most of the particles the relative error of the estimated position of the robot is less than 1%. The estimate coordinate is in [m], which has a greater error average with a value of 0.53%.

Map 3 results are presented in tables 5 and 6.

Table 5. General data map 3

Map size	256 m^2
Trajectory time	30 minutes
Distance traveled	245 m

Table 6. Errors obtained with map 3

Estimated position	**Relative error average (%)**	**Average absolute error**
Position X m	0.34	0.05 cm
Position Y m	0.53	0.07 cm
Orientation rad	0.43	0.001 rad

In map 3 evaluation, compared with maps 1 and 2, it is observed that decreases the peaks of errors in particles and also in the relative error of all coordinates estimated position, being the average absolute error which most decrease.

Considering errors magnitude at each position are in the range of between 1 [cm] to 12 [cm] and robot size has a dimension of 40 [cm] wide by 40 [cm] long, algorithms worked satisfactorily in map 3.

By comparing the evaluations graphs and tables for the 3 maps, it is observed that while minor is the space for the robot motion, lesser is the error in the estimation of the robot position when objects or walls have a uniform surface. On the other hand, when the robot explores in wider spaces or walls with a rough or discontinuous surface, the error is greater in the estimation of the robot position. This is due to that in tight spaces and plain, robot movements and the robot twists are more small and soft, on the contrary, in the open spaces and discontinuous, the robot movements and the robot twists are larger and rougher, causing the system of algorithms do not well estimate the position.

Using a real mobile robot, integrated algorithm is implemented with player stage set in the real robot, using a small netbook connected to the sensors and actuators. It is used the same parameters adjusted in simulations. The evaluation results are similar to simulation for map 1 and 2, with error closed to 1% in X position and 1.5% in Y position. With these results, it can say that SLAM integrated system with FastSLAM algorithms, field vectors histogram algorithm and spiral exploration algorithm, works with minimum errors in different evaluated environments.

5. Conclusions

The current paper has presented an integration of algorithms that permits maps construction and navigation of mobile robots. SLAM algorithm, based on FastSLAM method is used. Navigation system is based on VFH to avoid obstacle and a spiral way trajectory method. This allows autonomous indoor navigation for a mobile robot.

The map construction algorithm and robot locations are based on the method of FastSLAM. For autonomous exploration, system uses the VHF method to avoid obstacles and using spiral shape trajectory to explore the environment.

The developed system is evaluated by simulation in three different types of maps, with varying degrees of complexity in their structures, to know how algorithm works and to measurement errors in different environments evaluated. Then, player stage is set in the real robot, using a small netbook connected to the sensors and actuators.

Evaluations are made in simulation environment using Player/Stage. Then, player stage is set in the real robot, using a small netbook connected to the sensors and actuators.

Three scenarios are used for the simulation test: a map simple 1, 2 cave map, and a map 3, which is a maze. In the three maps simulated, in most of the particles the relative error of the estimated position of the robot is less than 1%. The estimate coordinate is in meter, with error average value of 0.53%. Errors are obtained in the estimation of the robot position produced by sudden movements of the robot and discontinuous surfaces on the map. These errors in the estimation of the integrated system can be reduced by decreasing the speed of the robot and modifying the parameters of the algorithms of the integrated system.

Using a real mobile robot, integrated algorithm is implemented with player stage set in the real robot, using a small netbook connected to the sensors and actuators. The same parameters adjusted in simulations have been used. The evaluation results are similar to simulation for map 1 and 2, with error closed to 1% in X position and 1.5% in Y position.

REFERENCES

1. HYUKDOO, C., W.-Y. KWANG, K. EUNTAI, **Simultaneous Global Localization and Mapping**, IEEE/ASME Transactions on Mechatronics, vol. 19(4), 2014, pp. 1160-1170.

2. HASHIKAWA, F., K. MORIOKA, **Mobile Robot Navigation based on Interactive SLAM with an Intelligent Space**. 8th International Conference on Ubiquitous Robots and Ambient Intelligence, 2011, pp. 788-789.

3. GARULLI, A., A. GIANNITRAPANI, A. ROSSI, A. VICINO, **Mobile Robots SLAM for Line-based Environment Representation**, Decision and Control Conference and European Control Conference, 2005.

4. LEE, Y., T. KWON, J. SONG, **SLAM of a Mobile Robot Using Thinning-based Topological Information**, International Journal of Control, Automation and Systems, vol. 5(5), 2007, pp. 577-583.

5. RESTREPO, C., A. ALZATE, C. TORRES, J. MOLINAS, **Localización y Construcción de un Mapa en Forma Simultánea Utilizando el Filtro Extendido de Kalman**, Revista de Ingeniería UC, Colombia, vol. 16(1), 2009, pp. 11-18.

6. OSORIO, R., S. GARCÍA, M. PENA, I. LOPEZ-JUAREZ, G. LEFRANC, **Movement and Color Detection of a Dynamic Object: An application to a Mobile Robot**, Studies in Informatics and Control, vol. 21, no. 1, 2012, pp. 33-40.

7. HADDA, I., J. KNANI, **Global Mapping and Localization for Mobile Robots Using Stereo Vision**, 10th International Multi-Conference on Systems, Signals & Devices, 2013, pp. 1-6.

8. MONTEMERLO, M., S. THRUN, **Simultaneous Localization and Mapping with Unknow Data Association Using FastSLAM**, Proceedings of the IEEE International Conference on Robotics Automation, USA, 2003, pp. 1985-1991.

9. MOUTTARLIER, P., R. CHATILA, **An Experimental System or Incremental Environment Modeling by an Autonomous Mobile Robot**, 1st International Symposium on Experimental Robotics, Canada, 2009.

10. HERNANDEZ-MARTINEZ, E. G., E. ARANDA-BRICAIRE, **Decentralized Formation Control of Multi-agent Robot Systems based on Formation Graphs**, Studies in Informatics and Control, vol. 21, no. 1, 2012, pp. 7-16.

11. FOX, D., **Adapting the Sample Size in Particle Filters Through KLD-Sampling**, International Journal of Robotics Research, vol. 22, no. 12, 2003, pp. 985-1003.

12. LATOMBE, J.-C., **Robot Motion Planning**, Kluwer Academic Publisher, 1991.

13. ULRICH, I., BORENSTEIN, J., **VHF+: Reliable Obstacle Avoidance for Fast Mobile Robots**, Proceedings of the IEEE International Conference on Robotics Automation, vol. 2, 1998, pp. 1572-1577.

14. MONTEMERLO, M., S. THRUN, D. KOLLER, B. WEGBREIT, **FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Probably Converges**, Proceedings of the Sixteenth International Conference on Artificial Intelligence, Mexico, 2003.

15. ROJAS, D., F. PASSOLD, C. CUBILLOS, G. LEFRANC, **Integration of Algorithms for Maps Construction and Simultaneous Localization in a Mobile Robot**, the 6th International Conference on Management and Control of Production and Logistics, Brazil, 2013.

16. FREES, D., C. CUBILLOS, G. LEFRANC, **Mobile Robot with Multi Agent Architecture**. IEEE International Conference on Engineering and Systems Applications, Santiago de Chile, 2012.