EXPERIMENTAL VALIDATION OF FASTSLAM ALGORITHM CHARACTERIZED BY A LINEAR FEATURES BASED MAP

Chiara Fulgenzi Gianluca Ippoliti Sauro Longhi

Dipartimento di Ingegneria Informatica,
Gestionale e dell'Automazione
Università Politecnica delle Marche
Via Brecce Bianche, 60131 Ancona, Italy
Email: chiaraful@gmail.com
{*gianluca.ippoliti, sauro.longhi*}*@univpm.it*

Abstract: In this paper the simultaneous localization and mapping problem in an unknown indoor environment is addressed. A probabilistic approach based on FastSLAM algorithm and line feature map is described. Experimental results performed with a smart wheelchair in an indoor environment are introduced and discussed.

Keywords: SLAM, FastSLAM, particle filters, line features

1. INTRODUCTION

The first and fundamental issue for every autonomous mobile robot is to be able to self-localize with respect to the surrounding environment. If a sufficiently precise map of the environment is available, the robot compares its observations with the map in order to understand its current position. In many practical applications however, the a priori knowledge of the environment is incomplete, uncertain, or not available at all: if this is the case, the robot should build the map during navigation and simultaneously, it uses the same map to compute its position. The problem is referred to as the *Simultaneous Localization And Mapping* (SLAM) problem and has gained a relevant position in robotics research since when (Smith and Cheeseman, 1986) gave a first closed formulation of the problem. The complexity of SLAM problem comes from the strong correlation between the localization and the mapping tasks: on one side, the robot pose has to be estimated with respect to the map; on the other one, the map itself is built with respect to the estimated robot pose. The problem has been deeply investigated in literature and different solutions have been proposed; for a comprehensive overview see (Thrun, 2002).

Many solutions are based on the Extended Kalman Filter (EKF) approach. The environmental map is usually represented by a list of features defined by appropriate parameters. The state vector describing the system is composed by both the state of the robot and the parameters of the map. At each time step new measures are acquired and the estimation of the state vector is updated using an EKF (Dyssanayake *et al.*, 2001; Ippoliti *et al.*, 2005; Bonci *et al.*, 2005). The convergence of the algorithm is assured under the linear gaussian hypothesis, when the position of at least one feature is known in advance and each feature is observed infinite number of times (Dyssanayake *et al.*, 2001). These assumptions are not always well fitted: the kinematic model of the robot is usually non linear and the uncertainty cannot in principle be assumed to be gaussian if there are ambiguities about which feature the measure is re-

ferred to (data association problem). Moreover the time required to update the full covariance matrix over the state vector scales quadratically with the number of features and real-time implementation in large environments becomes more and more expensive in terms of computational power.
Many other solutions are based on the particle filtering approach (Montemerlo *et al.*, 2002). The key idea of the proposed algorithm, called FastSLAM, is that if one could exactly know the robot trajectory, observations of different features would be statistically independent, then each feature in the map is updated separately. The uncertainty in robot pose is represented with a set of weighted samples, where each sample is a hypothesis of robot position and has attached his own map. Each feature is updated with an independent EKF attached to each particle. In such a way, the SLAM problem is decomposed in 1 problem of robot localization and N problems of feature position estimation, where N is the number of features in the environment. The non linear kinematic model can directly be considered, and different data association hypotheses can be maintained simultaneously. The complexity of the algorithm is proportional to the number of particles used and to the number of features. The algorithm has been proved to handle a large number of features in real time, unknown data association and loop closing problems (Nieto *et al.*, 2003; Hahnel *et al.*, 2003). The main contribution of this paper is to validate experimentally the FastSLAM algorithm in a completely unknown environment. A linear features based map is considered to simplify the environment representation. The experimental tests have been performed on a powered wheelchair equipped with a fiber optic gyroscope, a laser scanner and two optical encoders connected to the axes of the driving wheels. The obtained results are of interest in the emerging area of assistive technologies where powered wheelchairs can be used to strengthen the residual abilities of users with motor disabilities (Bourhis *et al.*, 2001; Fioretti *et al.*, 2000; Prassler *et al.*, 2001). The proposed approach results in a computationally efficient solution to the localization problem and may really represent a basic step towards the proper design of a navigation system aimed at enhancing the efficiency and the security of commercial powered wheelchairs.

Section 2 recalls Rao-Blackwellised particle filters and FastSLAM algorithm. Section 3 describes the sensor equipment used in our experiments and Section 4 describes the implementation details of the proposed solution to the SLAM problem. Finally, Section 5 shows the experimental results obtained on our mobile base. Discussion and comments end the paper.

2. THE FASTSLAM ALGORITHM

The FastSLAM algorithm here recalled, has been developed by (Montemerlo *et al.*, 2002). FastSLAM is an instance of the Rao-Blackwellized Particle Filter for the SLAM problem and in continuous state space approach. Particle filters can be regarded as a discrete approximation of more general Bayes filters. In a probabilistic framework, the uncertainty on the state of the system is explicitly considered through some parameterization of its Probability Density Function (PDFs). The key idea of particle filters is to approximate this function with sets of weighted samples, where each sample is an hypothesis on the state of the system and the attached weight is the probability that the real state is equal to the hypothesis (Doucet, 1998). Particle filters in robotics found a very successful application in the MonteCarlo localization algorithms (Dellaert *et al.*, 1999). Each particle represents a hypothesis on the robot pose and the weight is given by the likelihood of the external measure acquired. In the SLAM case however, the system is much more complex, as the state space is not simply described by the coordinates of the robot but also by all the parameters describing the map. Sampling efficiently in such a high dimensional space would require a huge number of samples (also called particles), so Rao-Blackwellized Particle Filters have to be considered (Murphy and Russel, 2000). The Rao-Blackwell rule allows to decompose the posterior PDF and to consider a partition of the state vector. The state space describing the robot pose is only sampled and updated by a particle filter while the state space describing the map is updated by an analytical filter conditional on each robot pose sample. The independence properties of the SLAM problem imply that, parameters referred to different map features can be updated independently one from the others. This means one has MN EKFs, where M is the number of particles (the dimension of the filter) and N is the number of features actually in the map. The state vector of each EKF is composed only by the parameters describing one feature. Following from these considerations, the FastSLAM algorithm is recalled (Montemerlo *et al.*, 2002):

0. Initialization: draw M independent identically distributed random samples from the PDF describing te initial robot pose;
1. Prediction: move samples according to the *probabilistic motion model*;
2. Update: acquire the new observation and weight the samples proportionally to likelihood of the measure;
3. Resampling step: normalize weights and resample particles in proportion to their weight;

4. Exact step: update the observed feature estimation of each resampled particle with an EKF. Go back to step 1.

Let $p(\mathbf{s}_0)$ be the PDF describing the knowledge about the position of the robot $\mathbf{s}_0$ at the beginning of the experiment. This function can be represented through a set of M identically distributed random samples (Doucet *et al.*, 2001):

$$s_0^i \sim p(\mathbf{s}_0)\ ,\ S_0 = \{\mathbf{s}_0^i\}_{i=1\ldots M}. \tag{1}$$

At each time instant k, the robot position s_k depends on the previous position s_{k-1} and on the known input controls u_k. In a probabilistic framework the uncertainty on the state of the robot can be described by the following PDF:

$$p(\mathbf{s}_k|\mathbf{s}_{k-1}^i, \mathbf{u}_k). \tag{2}$$

This function can be approximated moving the particles in $\mathbf{s}_{k-1}^i$, according to the *motion model*, which is usually a probabilistic generalization of the kinematic model and a new set of particles $\hat{S}_k = \{\hat{\mathbf{s}}_k^i\}_{i=1\ldots M}$ is obtained. Not hypotheses are made on the motion model, which can be for instance, non-linear and non-gaussian.
When the robot acquires a new measure of the environment, the new information is used to update the PDF of the robot pose and to decrease the uncertainty. The measure is compared with the expectation computed in each sample and the likelihood of the observation is used to weight the particles. To calculate the value of w_k^i of the weight of the particles, two sources of uncertainty on the measure have to be considered: the sensor noise and the error in the estimation on feature location. To take into a account the sensor noise, a *probabilistic observation model* is considered (Dellaert *et al.*, 1999):

$$p(\mathbf{z}_k|\mathbf{s}_k^i, \Pi^{j,i}), \tag{3}$$

which represents the PDF of the measurement $\mathbf{z}_k$ conditioned to the estimated robot pose $\mathbf{s}_k^i$ and to the expected location of the observed feature $\Pi^{j,i}$ and it is usually obtained by a probabilistic generalization of the measurement equation. To take into account the error in the estimation of feature location, the mean values and covariance of the feature are considered. In other words, the PDF of the feature localization conditioned to all the previous positions and measurement is considered $p(\Pi^{j,i}|\mathbf{s}^{i,k-1}, \mathbf{z}^{k-1})$ which takes into account the error in the feature position estimation. This PDF is always a gaussian with mean values vector and covariance matrix given by the EKF estimation at the previous time step. The value of the weight is then given by:

$$w_k^i \propto \int p(\mathbf{z}_k|\mathbf{s}_k^i, \Pi^{j,i}) \cdot p(\Pi^{j,i}|\mathbf{s}^{i,k-1}, \mathbf{z}^{k-1}) \cdot d\Pi^{j,i} \tag{4}$$

If both the first and second term are gaussians, the integral can be solved in closed form, and the weight can be computed.

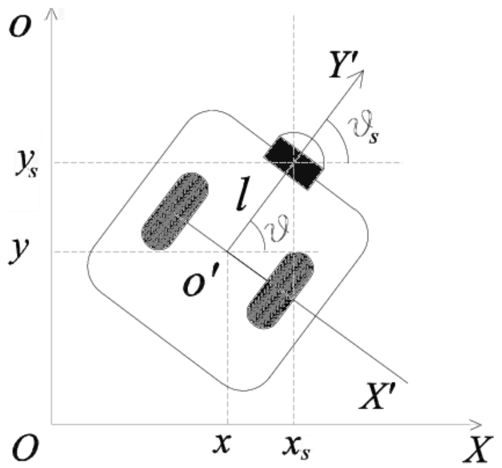


Fig. 1. Robot configuration.

Once all the M weights have been calculated, they are normalized by:

$$\tilde{w}_k^i = \frac{w_k^i}{\sum_{i=1}^{M} w_k^i} \tag{5}$$

and the particles are resampled. Each particle $\mathbf{s}_k^h$ in the new set S_k is the copy of a particle $\hat{\mathbf{s}}_k^i$ in $\hat{S}_k$, where:

$$P(\mathbf{s}_k^h = \hat{\mathbf{s}}_k^i) = \tilde{w}_k^i \tag{6}$$

Finally, also the map have to be updated. Each particle maintains an its own map, which is generally represented by a list of features each described by a vector of parameters (mean values) and a covariance matrix. Only the feature observed by the present measure is updated using a standard EKF; only the resampled particles have to be updated, as the others are discarded in the next iteration. To apply the EKF, the measurement model needs to be linearized and the noise should be approximated as gaussian.
In this description the data association problem is not considered. In Section 4 the proposed solution to this problem is presented.

3. THE SENSOR EQUIPMENT

3.1 Odometric measures

Consider an unicycle-like mobile robot with two driving wheels, mounted on the left and right sides of the robot, with their common axis passing through the center of the robot (see Figure 1). The pose of this mobile robot in a two-dimensional space can be uniquely identified by the coordinates x and y of the midpoint between the two driving wheels and the angle θ between the main axis of the robot and the X-direction. The kinematic model of the robot is described by the following equations in discrete time:

$$x(k+1) = x(k) + \Delta S \frac{sin\frac{\Delta\theta}{2}}{\frac{\Delta\theta}{2}} cos(\theta(k) + \frac{\Delta\theta}{2}) \quad (7)$$

$$y(k+1) = y(k) + \Delta S \frac{sin\frac{\Delta\theta}{2}}{\frac{\Delta\theta}{2}} sin(\theta(k) + \frac{\Delta\theta}{2}) \quad (8)$$

$$\theta(k+1) = \theta(k) + \Delta\theta. \quad (9)$$

where :

$$\Delta S = \frac{y_r(k) + y_l(k)}{2} r, \quad (10)$$

$$\Delta\theta = \frac{y_r(k) - y_l(k)}{d} r, \quad (11)$$

The terms $y_r(k)$ and $y_l(k)$ are the incremental distances covered on the time interval $(t(k), t(k+1)]$ by the right and left wheels of the robot, respectively. Denote by $\tilde{y}_r(k)$ and $\tilde{y}_l(k)$ the measures of $y_r(k)$ and $y_l(k)$ provided by the encoders:

$$\tilde{y}_r(k) = y_r(k) + n_r(k), \quad (12)$$

$$\tilde{y}_l(k) = y_l(k) + n_l(k), \quad (13)$$

where $n_r(\cdot)$, $n_l(\cdot)$ are the measurement errors, modeled as independent, zero mean, gaussian white sequences $(n_r(\cdot) \sim N(0, \sigma_r^2(k))$, $(n_l(\cdot) \sim N(0, \sigma_l^2(k))$. It follows that the really available values $\tilde{\Delta}S(k)$ and $\tilde{\Delta}\theta(k)$ of $\Delta S(k)$ and $\Delta\theta(k)$ are given by:

$$\tilde{\Delta}S(k) = \frac{\tilde{y}_r(k) + \tilde{y}_l(k)}{2} r = \Delta S(k) + n_S(k) \quad (14)$$

$$n_S(\cdot) \sim N\left(0, \sigma_S^2 = \frac{\sigma_r^2 + \sigma_l^2}{4} r^2\right) \quad (15)$$

$$\tilde{\Delta}\theta(k) = \frac{y_r(k) - y_l(k)}{d} r = \Delta\theta(k) + n_\theta(k) \quad (16)$$

$$n_\theta(\cdot) \sim N\left(0, \sigma_\theta^2 = \frac{\sigma_r^2 + \sigma_l^2}{d^2} r^2\right) \quad (17)$$

3.2 Fiber optic gyroscope measures

The operative principle of a Fiber Optic Gyroscope (FOG) is based on the Sagnac effect. The FOG is made of a fiber optic loop, fiber optic components, a photo-detector and a semiconductor laser. The phase difference of the two light beams traveling in opposite directions around the fiber optic loop is proportional to the rate of rotation of the fiber optic loop. A FOG does not require frequent maintenance, has a longer lifetime of the conventional mechanical gyroscopes, and has also a low drift. The FOG readings are denoted by:

$$\tilde{\Delta}\theta_g = \Delta\theta_g + n_{\theta_g} \quad (18)$$

$$n_{\theta_g}(\cdot) \sim N(0, \sigma_{\theta_g}^2) \quad (19)$$

where $\Delta\theta_g(\cdot)$ is the true value and $n_{\theta_g}(\cdot)$ is an independent, zero mean, gaussian white sequence.

As the FOG measure is much more reliable than the measure obtained from encoders readings $(\sigma_{\theta_g} << \sigma_\theta)$, we set $\tilde{\Delta}\theta = \tilde{\Delta}\theta_g$ to obtain a better estimation on robot motion.

3.3 Laser scanner measures

The distance readings by the Laser Measurement System (LMS) are related to the in-door environment model and the configuration of the mobile robot. Denote with l the distance between the center of the laser scanner and the origin O' of the coordinate system (O', X', Y') fixed to the mobile robot, as reported in Figure 1. At the sampling time k, the position of the center of the laser scanner, referred to the inertial coordinate system (O, X, Y), is:

$$\begin{bmatrix} x_s(k) \\ y_s(k) \\ \theta_s(k) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} l\cos\theta(k) \\ l\sin\theta(k) \\ 0 \end{bmatrix} \quad (20)$$

where l is defined in Figure 1. The walls and the obstacles in an semi-structured in-door environment can be represented by a proper set of planes orthogonal to the plane XY of the inertial coordinate frame. Each plane is represented by the vector $\Pi^j = [\rho^j, \alpha^j]^T$, where ρ^j is the normal distance of the plane from the origin O, α^j is the angle between the normal line to the plane and the X direction. In such a notation, the expectation of the i-*th* laser reading, $i = 1, 2, \cdots, n_l$, relative to the present distance of the center of the laser scanner from the plane Π^j, has the following expression (see Figure 2):

$$z_i^j(k) = G_i(\Pi^j, \mathbf{s}(k)) = = \frac{\rho^j - x_s(k)cos\alpha^j - y_s(k)sin\alpha^j}{cos\theta_i^j(k)} \quad (21)$$

where

$$\theta_i^j(k) = \alpha^j - (\theta_i - \frac{\pi}{2}) - \theta_s(k) \quad (22)$$

To compute an expectation of the measurement vector $\mathbf{z}(k) = [z_1, \ldots, z_{n_l}]^T$, one has to decide which feature is being observed from each laser beam (data association problem). In order to do this, the initial and final points of each line and possible occlusions have to be considered.
Although the LMS is a very accurate sensor, the noise in sensor readings has to be taken into account: the sensor noise is modeled with independent, zero-mean, gaussian white sequences on each element of the measurement vector:

$$\tilde{\mathbf{z}}(k) = \mathbf{z}(k) + [n_{z_1}, \ldots, n_{z_{n_l}}]^T, \quad (23)$$

$$\mathbf{n_z}(\cdot) \sim N(\mathbf{0}, R = diag\{\sigma_i^2\}_{i=1,\ldots,n_l}). \quad (24)$$

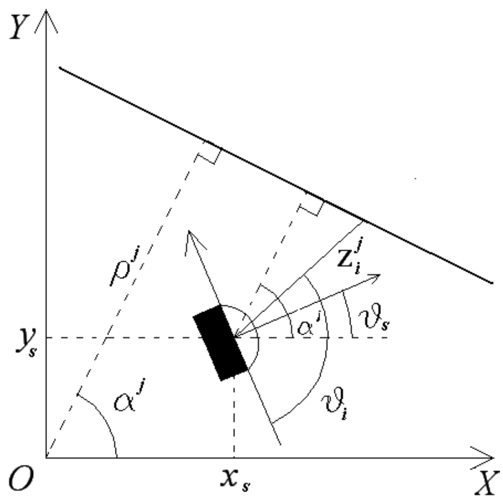


Fig. 2. Laser scanner measure.

4. DEVELOPED SOLUTION

4.1 Data association

Solving the data association problem in the SLAM case means not only to associate each measure with the structure that generated it, but also to know how and when to initialize new features. In the developed solution, both the problems are faced by a matching algorithm. A local map is built at each time step, i.e. each laser beam is associated to a line referred to the mobile reference frame or is discarded. This is done with a split-merge algorithm (Zhang and Ghosh, 2000): the measurement vector is recursively divided in clusters of nearby points and then in linear regions. Outliers and spurious measures are automatically deleted. For each region, the parameters ρ'^j and α'^j of the segment joining the first and last point of the same region are calculated. Now, local lines have to be compared with lines in the map for each particle: the parameters ρ_i^j and α_i^j referred to particle i are computed. Line j is said to match line g in the global map if:

1. the difference between α_j^i and α_g^i is less then a given threshold A;
2. the features can be partially superposed or are one next to the other.

Threshold A is chosen taking into account the noise of the laser sensor and the possible orientation error of the particle. To verify the second condition, the initial and final points of the features have to be considered. As the environment is initially unknown, the end points of a line at a given time step represent not an entire wall, but only the detected part of it, and they have to be updated each time a previously undetected part is observed. If both conditions are satisfied, then to all the measures referred to j are associated to g using a reference index and g is "stretched" to the new detected final/initial point. If j doesn't match any line in the map, then a minimum mean square algorithm is used to initialize mean values and covariance matrix of a new feature, while initial and final points are chosen to be the projection of the observed points on the computed line. The measures referred to new features will not be used to compute the particle weight (4) as it is not possible to compute the expectation.

4.2 SLAM algorithm

At the beginning of each experiment (k=0) the environment is totally unknown, and the configuration of the system for the available knowledge can be totally described by the robot pose $[0, 0, 0]^T$. When the robot acquires the first measure, the information extracted is stored in the initial map. Due to the sensor noise, this information is uncertain.
At k=1, the robot has moved according to the control inputs $\mathbf{u}_k = [\tilde{\Delta}S(k), \tilde{\Delta}\theta(k)]^T$, and the robot pose $\mathbf{s}_k = [x(k), y(k), \theta(k)]^T$ can be described by the PDF expressed by (2). A first set of M particles is drawn copying the initial position M times, $\{\mathbf{s}^i(0)\}_{i=1,\dots,M} = s_0$. In the next iterations ($k >= 2$) of the algorithm, particles $\mathbf{s}_{k-1}^i$ will instead be given by the resampling algorithm. The probabilistic motion model is obtained from the kinematic model and the statistic properties of the noise sequences (15) and (19), which are assumed to be known. In practice M random samples of noise are simulated, $\{n_S^i, n_\theta^i\}_{i=1,\dots,M}$ and M different control inputs $\mathbf{u}_k^i = [\tilde{\Delta}S^i(k), \tilde{\Delta}\theta^i(k)]^T$ are calculated. Applying these control inputs in the relations applying (7),(8) and (9), the new particles set $\{\hat{\mathbf{s}}^i(k)\}_{i=1,\dots,M}$ is obtained. As the model is non linear and non additive on gaussian noise, the resulting discrete distribution of particles is not necessarily gaussian.
When a new measure of the environment is acquired, the new information is used both to decrease the uncertainty on the robot pose and to add knowledge to the map. The first issue is faced weighting the particles with the likelihood w_k^i (4). The probabilistic observation model (3) is defined from the measurement equation (21) and the statistical properties of the sensor noise (24). Linearizing the observation model around the current state estimation the first term of the integral reported in (4) is gaussian. Moreover, updating features with an EKF, also the second term of the integral is gaussian and the weight can be calculated as follows:

$$w_i \propto \frac{\exp\left(-\frac{1}{2}\left(\mathbf{z} - \hat{\mathbf{z}}_i\right) Q_i^{-1} \left(\mathbf{z} - \hat{\mathbf{z}}_i\right)^T\right)}{(2\pi)^{\frac{d}{2}}\, Q_i^{\frac{d}{2}}}, \quad (25)$$

$$Q_i = \tilde{G}_i \cdot \Sigma_i^{-1} \cdot \tilde{G}_i^T + R, \quad (26)$$

where $\tilde{G}_i$ is the linearization of G function, Σ_i^{-1} is the inverse of the covariance matrix of Π^j estimated by particle i at the previous time step, $\hat{\mathbf{z}}_i$ is the expectation. In this notations, the time index k has been omitted. Once all the weights are computed, they are normalized by (5). The particle set for the next iteration of the algorithm is drawn from $\{\mathbf{s}^i(k)\}_{i=1,\dots,M}$ resampling particles with probability proportional to their weight with systematic resampling algorithm (Arulampalam *et al.*, 2002). In the general case, more than one line feature is observed at each time step, and, instead of doing a resampling step for each observed feature, as proposed in (Montemerlo *et al.*, 2002), the algorithm computes a total weight, given by the sum of single normalized weights. This is done in order to make the algorithm faster and more robust to particle *impoverishment*(Arulampalam *et al.*, 2002), that is when only one or a few particles are resampled, giving a bad estimate of the uncertainty.

Finally, the map of each resampled particle is updated using a standard EKF which state is Π^j for each observed feature j. To allow this, the sensor noise (24) has to be white, zero-mean and gaussian and the observation model (3) is linearized.

5. EXPERIMENTAL RESULTS

The experimental tests have been performed on the TGR Explorer powered wheelchair in an indoor environment. This vehicle has two driving wheels and a steering wheel. The odometric system is composed by two optical encoders connected to independent passive wheels aligned with the axes of the driving wheels. The incremental optical encoders SICOD mod. F3-1200-824-BZ-K-CV-01 have been used. This encoders have $1200\,pulses/rev.$ and a resolution of $0.0013\,rad$. The gyroscopic measures have been acquired in a digital form by a serial port on the on-board computer. The fiber optic gyroscope HITACHI mod. HOFG-1 have been used. The good estimate in orientation given by the optic gyroscope sensibly reduces the uncertainty of the robot pose and this fact allows to use a relatively low number of particles. In the performed experiments the number of particles is M=30. The laser scanner measures have been acquired by the SICK LMS mod. 200 installed on the vehicle. A resolution of 5 degrees and a spectrum of 180 degrees has been chosen in order to have a number of measures that could simultaneously guarantee good map building and real-time implementation. A sampling time of $0.4s$ has been used. The TGR Explorer powered wheelchair with data acquisition system for FOG sensor, incremental encoders and laser scanner is shown in Figure 3. Figures 4(a), 4(b), 5 and 6 illustrate samples of the obtained results. The smart wheelchair has been driven by the user interface in indoor environments of our Department. Markers have been put on the floor to measure the real trajectory of the wheelchair compared to the environment. A slight rotation had to be carried out in order to compare the map obtained by the algorithm and the real map of the environment. This is due to the uncertainty of the true initial position. Figure 4(a) shows the map built and the estimated position of the robot at time step $t = 80$. Figure 4(b) shows the trajectory of the vehicle and the map estimated by the proposed algorithm at the end of the experiment. The dots are the markers on the floor, the dot-dash lines represent the real map and the robot path estimated by the odometric and gyroscope measure only, while the solid lines are the robot pose and features estimation.

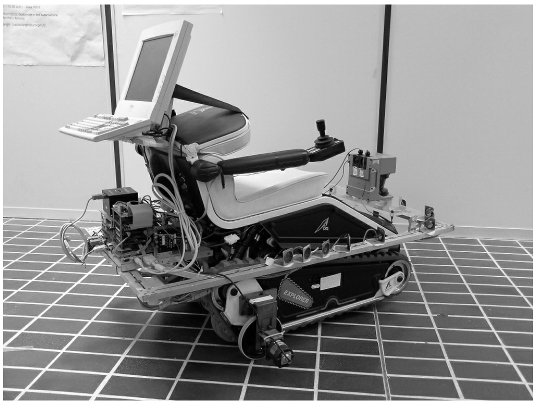


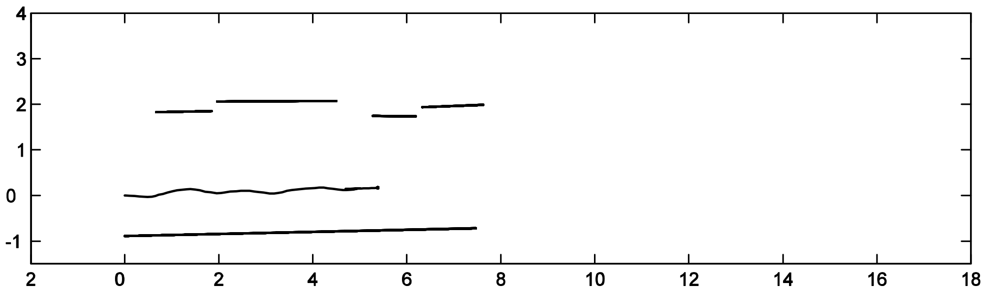
Fig. 3. TGR Explorer with data acquisition system for FOG sensor, incremental encoders and laser scanner.

Other experiments have been taken considering 130m long trajectories. The same number of particles, M=30, has been used. Figures 5 and 6 show the obtained results for two different environment configurations. The dots are the markers on the floor, the dot-dash lines represent the real map and the robot path estimated by the odometric and gyroscope measure only, while the solid lines are the robot pose and features estimation.

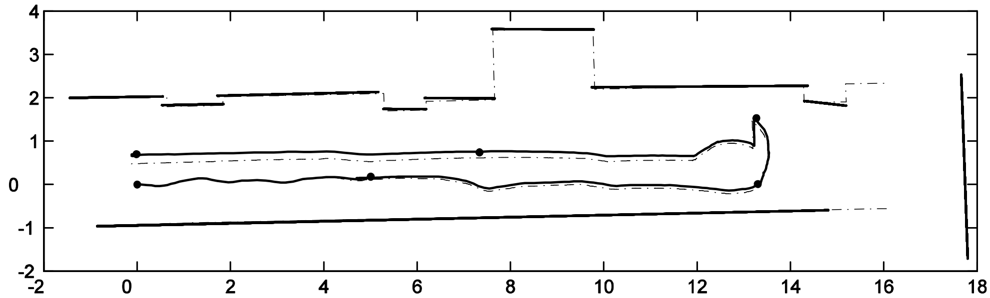
6. CONCLUDING REMARKS

In this paper, an implementation of the FastSLAM algorithm (Montemerlo *et al.*, 2002) with a line features based map is described. Some improvements have been introduced in order to reduce the number of particles and obtain a more robust algorithm. The implemented solution has been tested with some experiments in a completely unknown indoor environment.

Thanks to the gyroscope sensor accuracy, the number of particles needed to achieve good localization and mapping is reduced to few tens. Furthermore, the line features map gives a very



(a)



(b)

Fig. 4. (a) Estimated map and trajectory at $t = 80$; (b) Estimated VS real map and odometric trajectory. The unit of coordinate-axis is meter for both panels.

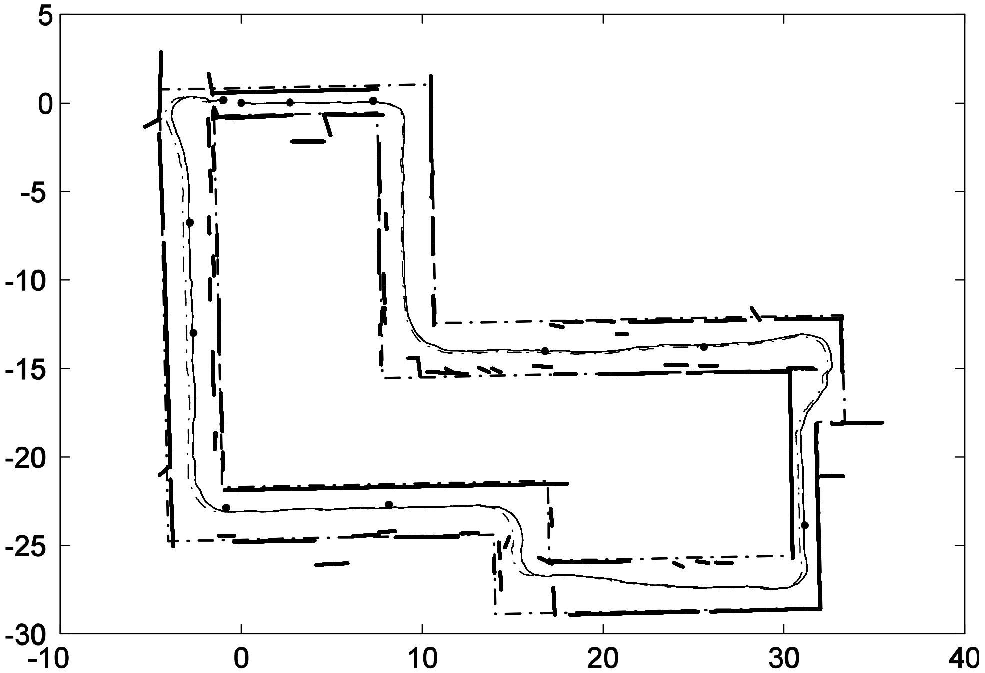


Fig. 5. Complete map of the Department corridors; 130m long trajectory. The unit of coordinate-axis is meter.

detailed and synthetic representation of the environment, and the mapping algorithm allows to estimate the measures referred to previously unseen parts of detected lines. The analyzed strategy is a computationally efficient solution to the mobile base localization in an unknown environment and it represents a suitable solution to develop a navigation system aimed at enhancing the efficiency and the security of the smart wheelchair.

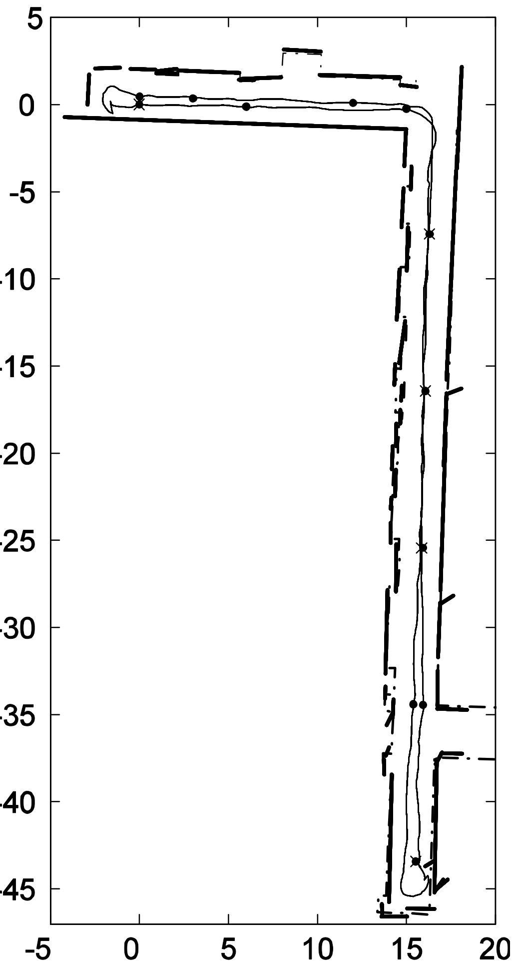


Fig. 6. Map of two Department corridors; 130m long trajectory. The unit of coordinate-axis is meter.

REFERENCES

Arulampalam, M.S., S. Maskell, N. Gordon and T. Clapp (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesiann tracking. *IEEE, Transaction on signal processing* **50**, NO.2.

Bonci, A., G. Ippoliti, A. La Manna, S. Longhi and L. Sartini (2005). Sonar and video data fusion for robot localization and environment feature estimation. *Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC05).*

Bourhis, G., O. Horn, O. Habert and A. Pruski (2001). An autonomous vehicle for people with motor disabilities. *IEEE Robotics and Automation Magazine* **7**(1), 20–28.

Dellaert, F., D. Fox, W. Burgard and S. Thrun (1999). Monte carlo localization for mobile robots. *Proc. IEEE Int. Conf. on Robotics and Automation* **1**, 1322–1328.

Doucet, A. (1998). On sequential simulation based methods for bayesian filtering. *Tecniocal report, Signal processing group, Department of engineering, University of Cambridge.*

Doucet, A., N. de Freitas and N. Gordon (2001). *Sequential Monte Carlo Methods in Practice.* Springer-Verlag.

Dyssanayake, G., P. Newman, S. Clark, H.F. Durrant-White and M.C. Sorba (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transaction on Robotics and Automation.*

Fioretti, S., T. Leo and S. Longhi (2000). A navigation system for increasing the autonomy and the security of powered wheelchairs. *IEEE Transactions on rehabilitation engineering* **8**(4), 490–498.

Hahnel, D., W. Burgard, D. Fox and S. Thrun (2003). An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurement. *Proc. IEEE Int. Conf. on Intelligent Robots and Systems* pp. 206–211.

Ippoliti, G., L. Jetto, A. La Manna and S. Longhi (2005). Improving the robustness properties of robot localization procedures with respect to environment features uncertainties. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005).* Barcelona, Spain. pp. 1463–1470.

Montemerlo, M., S. Thrun, D. Koller and B. Webright (2002). Fastslam: a factored solution to the simultaneous localization and mapping problem. *AAAI, Proceedings of the National conference on Artificial Intelligence.*

Murphy, K. and S. Russel (2000). Rao - blackwellised particle filtering for dynamic bayesian networks. *Proceedings of the UAI.*

Nieto, J., J. Guivant and E. Nebot (2003). Real time data association for fastslam. *Proc. IEEE Int. Conf. on Robotics and Automation* pp. 412–418.

Prassler, E., J. Scholz and P. Fiorini (2001). A robotic wheelchair for crowded public environments. *IEEE Robotics and Automation Magazine* **7**(1), 38–45.

Smith, R.C. and P. Cheeseman (1986). On the representation and estimation of partial uncertainty. *Int. journal of Robotics research* **5**, 56–68.

Thrun, S. (2002). Mapping, a survey. *IEEE proceedings* **1**, 123–126.

Zhang, L. and B.K. Ghosh (2000). Line segment based map building and localization using 2d laser rangefinder. *Proc. IEEE Int. Conf. on Robotics and Automation* pp. 2538–2543.