Experimental validation of FastSLAM algorithm integrated with a linear features based map

Chiara Fulgenzi, Gianluca Ippoliti, Sauro Longhi

▶ **To cite this version:**

HAL Id: inria-00360211

https://hal.inria.fr/inria-00360211

Submitted on 10 Feb 2009

Experimental validation of FastSLAM algorithm integrated with a linear features based map

Chiara Fulgenzi [a], Gianluca Ippoliti [b], Sauro Longhi [b],*

[a]*INRIA Rhône-Alpes, LIG, Grenoble, France*
Email: chiara.fulgenzi@inriaples.fr

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

Abstract

In this paper the Simultaneous Localization And Mapping (SLAM) problem in unknown indoor environments is addressed. A probabilistic approach integrating FastSLAM algorithm and a line feature map is developed and validated. Experimental validation is performed by a smart wheelchair equipped with proprioceptive and exteroceptive sensors in an office like environment where loop closing is achieved without any dedicated algorithm. Geometric hypothesis of orthogonal line features are considered to enhance the performance of the algorithm in the considered environment. The proposed approach results in a computationally efficient solution to the SLAM problem and the high quality sensor measurements allow to maintain a good localization of the mobile base and a compact representation of the environment.

Key words: SLAM, FastSLAM, particle filters, line features

* Tel.: +39 071 2204451; fax: +39 071 220 4474.
  Email address: `sauro.longhi@univpm.it` (Sauro Longhi).

Nomenclature

M	Number of particles
N	Number of features in the map
$\mathbf{s}_k$	Pose of the robot at time instant k
$x(k)$	x coordinate of the robot at time instant k
$y(k)$	y coordinate of the robot at time instant k
$\theta(k)$	Orientation of the robot at time instant k
$\mathbf{u}_k$	Input controls vector at time instant k
v	Linear velocity of the robot
ω	Angular velocity of the robot
$\mathbf{z}_k$	Measurement vector at time instant k
$\mathbf{s}_k^m$	Particle m at time instant k
w_k^m	Weight of particle m at time instant k
$\tilde{w}_k^m$	Normalized weight of particle m at time instant k
$\mathcal{S}_k$	Particle set at time k
$\hat{\mathcal{S}}_k$	Particle set before resampling step at time instant k
σ_s	Standard deviation of distance measurement noise
σ_θ	Standard deviation of orientation measurement noise
$\mathbf{\Pi}^m$	Map of m-th particle, list of features
$\mathbf{\Pi}_j$	Parameter vector of feature j
$\mathbf{\Pi}_j^m$	Parameter vector of feature j estimated by the m-th particle
ρ_j	Normal distance between feature j and the origin of the absolute reference frame
α_j	Angle between the normal to feature j and the X direction of the absolute reference frame
ϵ	Angular threshold for features association
D	Distance threshold for features association
$(\cdot)^k$	Indicates variable set from initial time instant 0 to k

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, 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 should use 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 a first closed formulation of the problem was

introduced in [1]. 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 [2–4]. Many solutions are based on the Extended Kalman Filter (EKF) approach. The environment 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 system state vector is updated using an EKF [5–9]. 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 [6]. These assumptions are not always verified: 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 the feature which is related to the measure (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.

To overcome to these problems, various approaches based on Rao-Blackwellised particle filters have been developed. In particular, the key idea of the Fast-SLAM algorithms [10], 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 trajectory and has attached its 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 [11,12]. Other particle filter based algorithms have been developed in order to handle occupancy grids maps [13]. The advantage of using such maps is that is not necessary to extract geometrical features from the raw sensor data, nor to make hypotheses on the geometry of the surrounding environment, which make the algorithms suitable for both indoor and outdoor environments. On the other side the memory requirement and the computation effort needed scales with each dimension of the environment. Further improvement is than

3

necessary in order to satisfy real-time constraints [13].

The main contribution of this paper is to validate experimentally the Fast-SLAM algorithm in a completely unknown indoor environment. An accurate sensor model is presented and a linear features based map is used. Such a representation leads to many advantages in an office like environment as shown by experimental results: the map is compact and adequate to model the real structure of the environment; the robot maintains good localization along the path, which allows not only to reduce the set of particles that represent localization uncertainty but also to close loops without any dedicated algorithm. The hypothesis of orthogonal features has been also assumed to enhance the performance of the algorithm in the considered indoor environment. The use of this algorithm in a more general environment, for example an outdoor environment, requires a more extended set of environment primitives.

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 high quality of inertial data acquired allows to further reduce the number of particles needed for the algorithm to converge. 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 [14–18]. The proposed approach results in a computationally efficient solution to the SLAM 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 the performed experiments and Section 4 describes the implementation details of the proposed solution to the SLAM problem. Finally, Section 5 shows the experimental results obtained by the considered mobile base. Discussion and comments end the paper.

2 The FastSLAM algorithm

The FastSLAM algorithm here recalled, has been developed in [10]. FastSLAM is an instance of the Rao-Blackwellized Particle Filter for a state space approach to the SLAM problem. Particle filters can be considered 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 (PDF). 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 [19]. Particle filters in robotics found a very successful application in the Monte-Carlo localization algorithms [20]. 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. In such a high dimensional space a huge number of samples (also called particles) are required, therefore Rao-Blackwellized Particle Filters have been introduced [21]. 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 and hence a number of MN EKFs are necessary, where M is the number of particles 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. In the following, the FastSLAM algorithm proposed in [10] is recalled and some remarks are also introduced:

0. Initialization: draw M independent identically distributed random samples from the PDF describing the 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 [22]:

$$\mathbf{s}_0^m \sim p(\mathbf{s}_0) \ , \ \mathcal{S}_0 = \{\mathbf{s}_0^m\}_{m=1\ldots M}. \tag{1}$$

At each time instant k, the robot position $\mathbf{s}_k$ depends on the previous position $\mathbf{s}_{k-1}$ and on the known input controls $\mathbf{u}_k$.

In a probabilistic framework the uncertainty on the state of the robot can be described by the following PDF:

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

This function can be approximated moving the particles in $\mathbf{s}_{k-1}^m$, according to the *motion model*, which is usually a probabilistic generalization of the robot kinematic model and a new set of particles $\hat{\mathcal{S}}_k = \{\hat{\mathbf{s}}_k^m\}_{m=1\ldots M}$ 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 w_k^m 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 [20]:

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

which represents the PDF of the measurement $\mathbf{z}_k$ conditioned to the robot pose $\mathbf{s}_k^m$ and to the location of the observed feature $\mathbf{\Pi}_j^m$ and it is usually obtained by a probabilistic generalization of the measurement equation. The estimation of the feature localization conditioned to all the previous positions and measurements $p(\mathbf{\Pi}_j^m | \mathbf{s}^{m,k-1}, \mathbf{z}^{k-1})$ is considered. This PDF represent the uncertainty in the feature position estimation and is always a Gaussian with mean value vector and covariance matrix given by the EKF recursion at the previous time step. The value of the weight is then given by:

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

If both the first and second terms are Gaussian, the integral can be solved in closed form, and the weight can be computed. Once all the M weights have been calculated, they are normalized and the particles are resampled. Each particle $\mathbf{s}_k^h$ in the new set $\mathcal{S}k$ is the copy of a particle $\hat{\mathbf{s}}_k^m \in \hat{\mathcal{S}}_k$, where:

$$P(\mathbf{s}_k^h = \hat{\mathbf{s}}_k^m) = \tilde{w}_k^m \tag{5}$$

and $\tilde{w}_k^m$ is the normalized weight. Finally, also the map has to be updated. The map is the list of the observed features, each described by a vector of parameters (mean values) and a covariance matrix. Each particle maintains its own map and only the features 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 is approximated with a 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 Fiber optic gyroscope measures

The Fiber Optic Gyroscope (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 travelling in opposite directions around the fiber optic loop is proportional to the rate of rotation (Sagnac effect). 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 :

$$\tilde{\Delta}\theta = \Delta\theta + n_g \tag{6}$$

where $\Delta\theta(\cdot)$ is the true value and $n_g(\cdot)$ is an independent, zero mean, Gaussian white sequence with standard deviation σ_θ [23–27]. As the FOG measure is much more reliable than the measure obtained from encoders readings it is used for robot pose estimation and it is assumed that the estimation of the robot angular velocity ω coincides with $\tilde{\Delta}\theta$, i.e. $\omega = \tilde{\Delta}\theta$. Moreover, encoders readings are used for estimation of the linear robot velocity v making use of standard odometric equations [28].

3.2 Laser scanner measures

The distance readings by the Laser Measurement System (LMS) are related to the indoor environment model and the configuration of the mobile robot [29,30]. At the sampling time k, the position of the center of the laser scanner,
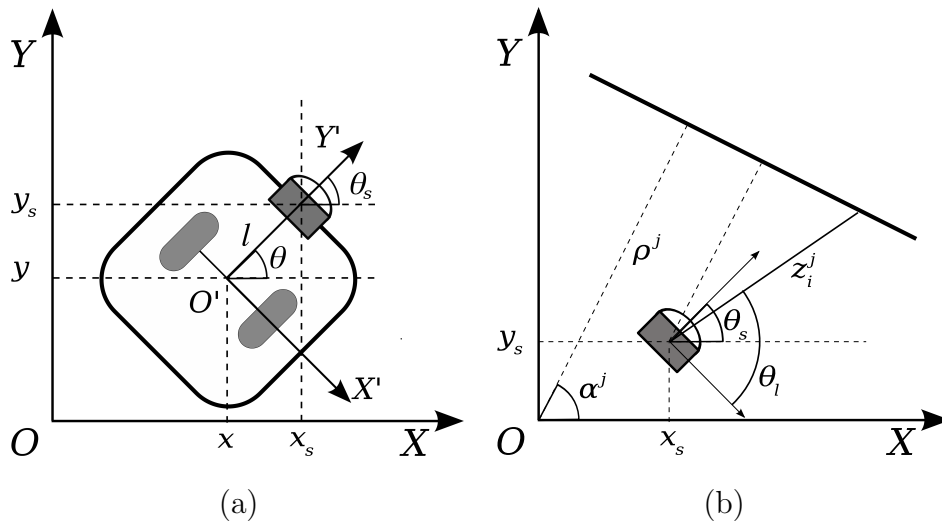

Fig. 1. (a) Robot Configuration, (b) Laser scanner measure.

referred to the inertial coordinate system (O, X, Y), is:

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

as shown in Figure 1.

The walls and the obstacles in a semi-structured indoor environment can be represented by a proper set of planes orthogonal to the ground. Each plane is represented by the vector $\mathbf{\Pi}_j = [\rho_j, \alpha_j]^T$, 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, $i = 1, \cdots, n_l$, relative to the plane $\mathbf{\Pi}_j$, has the following expression (see Figure1):

$$z_i^j(k) = G_i(\mathbf{\Pi}_j, \mathbf{s}(k)) = \frac{\rho_j - x_s(k)cos(\alpha_j) - y_s(k)sin(\alpha_j)}{cos(\theta_i^j(k))}, \qquad (7)$$

where:

$$\theta_i^j(k) = \alpha_j - \left(\theta_i - \frac{\pi}{2}\right) - \theta_s(k). \qquad (8)$$

To compute an expectation of the measurement vector $\mathbf{z}_k = [z_1(k), \ldots, z_{n_l}(k)]^T$, one has to decide which feature is being observed from each laser beam. In order to do this, the initial and final points of each line and possible occlusions have to be considered (data association problem: see Subsection 4.1). 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 [29,30].

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 [31]: the measurement vector is recursively divided in clusters of nearby points

and then in linear regions. Outliers and spurious measures are automatically discarded. 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^m_j and α^m_j referred to particle m are computed. Line j is said to match line g in the global map if:

1. the difference between α^m_j and α^m_g is less then a given threshold ϵ;

2. the features can be partially superposed or are one next to the other, according to threshold D.

Threshold ϵ is chosen taking into account the noise of the laser sensor and the possible orientation error of the particle; threshold D takes into account the accuracy of the sensor and the desired precision of the map. 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 do not represent 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 all the measures referred to line j are associated to line g using a reference index (Algorithm 1). Successively g is "stretched" to the new detected final/initial point (Algorithm 2). If line 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 as the projection of the observed points on the computed line. The measures referred to new features are not be used to compute the particle weight (equation 4) as it is not possible to compute the expectation.

The newly observed features will be then added to the global map. The map in each particle is maintained ordered: at each time step, the first features checked for matching are the ones observed at the previous step.

Algorithm 1: Data Association

Data: $\mathbf{\Pi}$, $\mathbf{\hat{\Pi}}$
Result: $\mathcal{H}$
1   $\mathcal{H} = -1$;
2   $g = 0$;
3   **for** $(j = 1$ **to** length $(\mathbf{\Pi}))$ **do**
4   retrieve α_j, ρ_j from $\mathbf{\Pi}_j$;
5   **while** $(g < \text{length}(\mathbf{\hat{\Pi}})$, $\alpha_g < \alpha_j)$ **do**
6   retrieve α_g and ρ_g from $\mathbf{\hat{\Pi}}_g$;
7          $g + +$;
8   **end**
9   **while** $(\mathcal{H}_j = -1$, $g \leq \text{length}(hat\mathbf{\Pi})$, $|\alpha_g - \alpha_j| < \epsilon)$ **do**
10          $d = \text{computeDistance} (\mathbf{\Pi}_j, \mathbf{\hat{\Pi}}_g)$;
11   **if** $d \leq D$ **then** $\mathcal{H}_j = g$;
12          $g + +$;
13   retrieve α_g, ρ_g from $\mathbf{\hat{\Pi}}_g$;
14   **end**
15   **end**

Algorithm 2: UpdateMap1

Data: $(\Pi, \hat{\Pi}, \mathcal{H})$

Result: $\hat{\Pi}$, $NewFeatures$

1 $NewFeatures = \emptyset$;
2 **for** $(j = 1$ **to** $\text{length}(\Pi))$ **do**
3   retrive X_{j1}, X_{j2}, extreme points of Π_j;
4      $g = \mathcal{H}_j$;
5   **if** $g \geq 0$ **then**
6   retrive ρ_g, α_g and X_{g1}, X_{g2} extreme points of $\hat{\Pi}_g$;
7          $d[1] = \text{ComputeDistance}(X_{l1}, X_{g1})$;
8          $d[2] = \text{ComputeDistance}(X_{l1}, X_{g2})$;
9          $d[3] = \text{ComputeDistance}(X_{l2}, X_{g1})$;
10         $d[4] = \text{ComputeDistance}(X_{l2}, X_{g2})$;
11         $i = \text{IndexOfMax}(d)$;
12   **switch** (i) **do**
13   **case** 1 $X_{g2} = X_{j1}$;
14   **case** 2 $X_{g1} = X_{j1}$;
15   **case** 3 $X_{g2} = X_{j2}$;
16   **case** 4 $X_{g1} = X_{j2}$;
17   **end**
18   take the orthogonal projection of X_{g1}, X_{g2} to ρ_g, α_g as new extreme points of $\hat{\Pi}_g$;
19   **end**
20   **else** $NewFeatures = \text{add}(\Pi_j)$;
21 **end**

4.2 SLAM algorithm

At the beginning of each experiment $(k = 0)$ the environment is totally unknown, and the configuration of the system 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. At $k = 1$, the robot has moved according to the control inputs $\mathbf{u}_k = [v(k), \omega(k)]^T$ and the robot pose $\mathbf{s}_k = [x(k), y(k), \theta(k)]^T$ can be described by the PDF expressed by Equation (2). A first set of M particles is drawn copying the initial position M times, $\{\mathbf{s}_0^m\}_{m=1,\dots,M} = \mathcal{S}_0$. In the next iterations $(k \geq 2)$ of the algorithm, particles $\mathbf{s}_{k-1}^m$ will instead be given by the resampling algorithm. The probabilistic motion model is obtained from the robot kinematic model and the statistic properties of the noise sequences σ_s and σ_θ, which are assumed to be known. In practice M random samples of noise are simulated, $\{n_v^m, n_\omega^m\}_{m=1,\dots,M}$ and M different control inputs $\mathbf{u}_k^m = [v^m(k), \omega^m(k)]^T$ are used. The new particles set $\hat{\mathcal{S}}_k$ is obtained. As the model is non linear and non additive on Gaussian noise, the resulting discrete distribution of particles is not necessarily Gaussian. At each new measure, the 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^m of Equation (4). The probabilistic observation model (3) is defined from the measurement Equation (7) and the statistical properties of the sensor noise. Linearizing the observation model around the current state estimation and updating features with an EKF, both the terms of the integral reported in Equation (4) are Gaussian and the weight

can be calculated as follows, where the time index k has been omitted:

$$w^m \propto \frac{\exp\left(-\frac{1}{2}\left(\mathbf{z} - \hat{\mathbf{z}}^m\right)(Q^m)^{-1}\left(\mathbf{z} - \hat{\mathbf{z}}^m\right)^T\right)}{(2\pi)^{\frac{d}{2}}(Q^m)^{\frac{d}{2}}}, \tag{9}$$

$$Q^m = \tilde{G}^m \cdot (\Sigma^m)^{-1} \cdot (\tilde{G}^m)^T + R, \tag{10}$$

where $\tilde{G}^m$ is the linearization of G function, $(\Sigma^m)^{-1}$ is the inverse of the covariance matrix of $\mathbf{\Pi}_j$ estimated by particle m at the previous time step, $\hat{\mathbf{z}}^m$ is the expectation. All the weights are computed and normalized. The particle set for the next iteration of the algorithm is drawn from $\mathcal{S}_k$ resampling particles with probability proportional to their weight with systematic resampling algorithm [32]. 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 [10], the algorithm computes a total weight, given by the sum of single normalized weights. This makes the algorithm faster and more robust to particle *impoverishment* [32]. Finally, the map of each resampled particle is updated using a standard EKF which state is $\mathbf{\Pi}_j$ for each observed feature j. To allow this, the sensor noise has to be white, zero-mean and Gaussian and the observation model (3) is linearized. Algorithm 3, along with Algorithms 1 and 2, describes the implementation of the proposed method. Consider-

---

Algorithm 3: FastSLAM

1   FastSLAM($\mathcal{S}_{k-1}$)
2   $\mathbf{z}_k = acquireLaser()$;
3   $\mathbf{u}_k = acquireOdometry()$;
4   $\pi_k = \text{splitAndMerge}(\mathbf{z}_k)$;
5   **for** $m = 1$ **to** M **do**
6      $\mathbf{s}_k^m = \text{DrawFromMotionModel}(\mathbf{s}_{k-1}^m, \hat{\mathbf{u}}_k)$;
7      $\mathbf{\Pi}_k^m = \text{LocalToGlobalCartesianFrame}(\pi_k, \mathbf{s}_k^m)$;
8      $\mathcal{H}_k^m = \text{DataAssociation}(\mathbf{\Pi}_k^m, \hat{\mathbf{\Pi}}_{k-1}^m)$;
9      $[\hat{\mathbf{\Pi}}_k^m, NewFatures^m] = \text{UpdateMap1}(\mathbf{\Pi}_k^m, \hat{\mathbf{\Pi}}_{k-1}^m, \mathcal{H}_k^m)$;
10   **for** $j = 0$ **to** $j = \text{length}\,\mathbf{\Pi}$ **do**
11         $g = \mathcal{H}_k^m(j)$;
12         $w_k^m = w_k^m + likelihood(\mathbf{\Pi}_{j,k}^m, \hat{\mathbf{\Pi}}_{g,k}^m)$;
13   **end**
14   **end**
15   $\mathbf{W_k} = \text{Normalize}(\mathbf{W_k})$;
16   Resample($\mathbf{W_k}$);
17   **for** $m = 1$ **to** M **do**
18   **if** m is sampled **then**
19         $\hat{\mathbf{\Pi}}_k^m = \text{UpdateMap2}(\mathbf{\Pi}_k^m, \hat{\mathbf{\Pi}}_k^m, \mathcal{H}_k^m)$;
20         $\hat{\mathbf{\Pi}}_k^m = \text{add}(NewFeatures^m)$;
21   **end**
22   **for** $m = 1$ **to** M **do**
23      $\mathcal{S}_k^m = [\mathbf{s}_k^m, \hat{\mathbf{\Pi}}_k^m]$;
24   **end**
25   **return** $\mathcal{S}_k$;

---

ing only line features is a well suited hypothesis in office-like environments (see also [33]). Navigation in an outdoor environment instead, requires a more general algorithm and a more complex setting. For example, in [34] a set of

different primitives is used. In this paper the achieved result is a simple and compact map, useful for navigation in office like environments, in which the robot and the user can easily recognize corridors and openings (doors). In the case of an office-like environment also, it is very likely that the walls are parallel or orthogonal one to the others. Introducing this hypothesis in the considered algorithm would lead to a faster computation without significantly affect the precision of the map of the considered environment. Algorithm 3 has been modified to handle this hypothesis. At the beginning a reference line is chosen from the scan and its angle α_0 is considered as the reference angle. All the new features are forced to be orthogonal ($\alpha = \alpha_0 + \frac{\pi}{2}$, $\alpha = \alpha_0 + \frac{3\pi}{2}$) or parallel ($\alpha = \alpha_0$, $\alpha = \alpha_0 + \pi$) to the reference. The same approach is used after each EKF estimation. The feature parameter ρ and the initial and final points have also to be corrected maintaining the medium point of the line fixed. This procedure is detailed in Algorithm 4.

Algorithm 4: OrthogonalMap

Data: $(\hat{\mathbf{\Pi}}, \alpha_0)$
Result: $\hat{\mathbf{\Pi}}$

1 **for** $(g = 0 \textbf{ to } \text{length}(\hat{\mathbf{\Pi}}))$ **do**
2   retrieve $\langle \alpha_g, \rho_g, X_{g1}, X_{g2} \rangle$;
3   **if** $|\alpha_g - \alpha_0| \leq \frac{\pi}{4}$ **then** $\alpha_g = \alpha_0$;
4   **else if** $|\alpha_g - \alpha_0| \leq \frac{3\pi}{4}$ **then** $\alpha_g = \alpha_0 + \frac{\pi}{2}$;
5   **else if** $|\alpha_g - \alpha_0| \leq \frac{5\pi}{4}$ **then** $\alpha_g = \alpha_0 + \pi$;
6   **else if** $|\alpha_g - \alpha_0| \leq \frac{7\pi}{4}$ **then** $\alpha_g = \alpha_0 + \frac{3\pi}{2}$;
7   **else** $\alpha_g = \alpha_0$;
8      $\bar{x}_g = (X_{g1}.x + X_{g2}.x)/2$;
9      $\bar{y}_g = (X_{g1}.y + X_{g2}.y)/2$;
10      $\rho_g = \bar{x}_g * cos(\alpha_g) + \bar{y}_g * sin(\alpha_g)$;
11   take the orthogonal projection of X_{g1}, X_{g2} to ρ_g, α_g as new extreme points of $\hat{\mathbf{\Pi}}_g$;
12 **end**

---

5   Experimental results

The experimental tests have been performed on the TGR Explorer powered wheelchair [35] 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, as shown in Figure 2. The odometric data are the incremental measures that at each sampling interval are provided by the encoders. The incremental optical encoders SICOD mod. F3-1200-824-BZ-K-CV-01 have been used [36]. Each encoder has $1200\, pulses/rev.$ and a resolution of $0.0013\, rad.$ These measures are directly acquired by the low level controller of the mobile base. The gyroscopic measures have been acquired in a digital form by a serial port on the on-board computer. The Fiber Optic Gyroscope (FOG) HITACHI mod. HOFG-1 has been used [37]. The good estimate of the vehicle orientation sensibly reduces the uncertainty of the robot pose and this fact allows the use
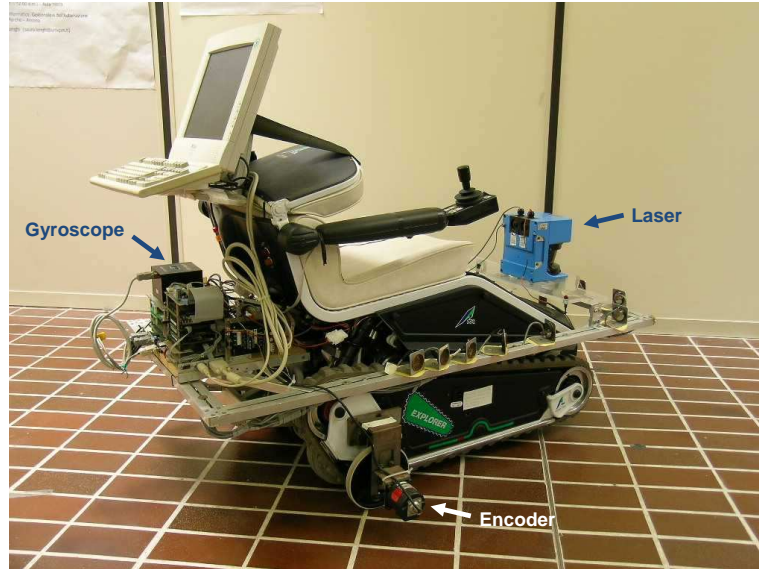
Fig. 2. TGR Explorer with data acquisition system for FOG sensor, incremental encoders and laser scanner.

of a reduced number of particles. The main characteristics of the FOG are reported in Table 1. While the used FOG measures rotational rates with a very high accuracy, the internal integration to derive the heading angle can suffer from drift [25,26]. Because of the low rate integration drift of the used FOG (see Table 1), the drift is not accounted for in the proposed experiments where the robot task duration is on the order of several minutes. For longer task duration the rate integration drift can be compensated as proposed in [27] or can be periodically reset by a proper docking system or an absolute sensing mechanism [25]. The laser scanner measures have been acquired by the

Table 1
Characteristics of the HITACHI gyroscope mod. HOFG - 1

Rotation Rate	-1.0472 to $+1.0472\,rad/s$
Angle Measurement Range	-6.2832 to $+6.2832\,rad$
Random Walk	$\leq 0.0018\,rad/\sqrt{h}$
Zero Drift (Rate Integration)	$\leq 0.0175\,rad/h$
Non-linearity of Scale Factor	within $\pm 1.0\%$
Time Constant	Typ. $0.02\,s$
Response Time	Typ. $0.02\,s$
Data Output Interval	Min. $0.01\,s$
Warm-up Time	Typ. $6.0\,s$

SICK LMS mod. 200 installed on the vehicle [38]. A resolution of 0.5 degrees and a spectrum of 180 degrees has been chosen.

A sampling time of 0.4s has been used. The main characteristics of the LMS

are reported in Table 2. A characterization study of the Sick LMS 200 laser scanner has been performed as proposed in [29]. Different experiments have been carried out to analyse the effects of data transfer rate, drift, optical properties of the target surfaces and incidence angle of the laser beam. Based on empirical data a mathematical model of the scanner errors has been obtained. This model has been used as a calibration function to reduce measurement errors. The TGR Explorer powered wheelchair with data acquisition system

Table 2
Characteristics of the SICK laser scanner mod. 200

Aperture Angle	$3.14\,rad$
Angular Resolution	$0.0175/\,0.0088/\,0.0044\,rad$
Response Time	$0.013/\,0.026/\,0.053\,s$
Resolution	$0.010\,m$
Systematic Error	$\pm 0.015\,m$
Statistic Error (1 Sigma)	$0.005\,m$
Laser Class	1
Max. Distance	$80\,m$
Transfer Rate	$9.6/\,19.2/\,38.4/\,500\,kBaud$

for FOG sensor, incremental encoders and laser scanner is shown in Figure 2. The FastSLAM algorithm has been implemented on the navigation module of the TGR Explorer powered wheelchair [35]. The navigation module developed for the considered vehicle interacts with the user in order to involve her/him in the guidance of the vehicle without limiting the functionality and the security of the system. The user sends commands to the navigation module through the user interface and the module translates the user commands in the low level command for the driving wheels. Two autonomy levels are developed to perform a simple filtering or to introduce some local corrections of the user commands on the basis of vehicle pose and environment information estimated by the considered FastSLAM algorithm (for more details see [17]). The navigation system is connected directly with the low level controller, the fiber optic gyroscope and the laser scanner by analog and digital converters and serial ports RS232, respectively.

Figures 3, 4, 5, 6 illustrate samples of the developed experiments. The parameters used respectively for the first experiment (Figures 3, 4 and 5) and for the second (Figure 6) are resumed in Table 3. The smart wheelchair has been driven by the user interface in indoor environments of the Dipartimento di Ingegneria Informatica, Gestionale e dell' Automazione. 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 with respect to the

14

Table 3
Parameters used in the experiments

| Parameter Name	Exp. 1	Exp. 2
M	30	30
σ_s	0.07	0.07
σ_θ	0.0001	0.05
ϵ	$\frac{\pi}{10}$	$\frac{\pi}{6}$
D	$0.15m$	$0.15m$

real environment. Figure 3(a) shows the real map (green lines), the map built by the FastSLAM algorithm (black lines) and the estimated position of the robot (red line) at time step $t = 80s$. Figure 3(b) shows the trajectory of the vehicle and the map estimated by the proposed algorithm at the end of the experiment. In Figure 3(b) blue dots are the markers on the floor, green lines represent the real map, the blue line is the robot path estimated by the odometric and gyroscope measure only, while black lines are the robot features estimation obtained by the FastSLAM algorithm; the red line is the trajectory estimated by the algorithm. Other experiments have been taken considering 130m long trajectories. The same number of particles, $M = 30$, has been used. Figures 4 and 5 show the obtained results for two different environment configurations. Blue dots are the markers on the floor and green lines represent the real map; the black lines and the red trajectory are estimations obtained by the considered FastSLAM algorithm.

To better evaluate the performance of the proposed algorithm, the following two tests excluding the high precision gyroscope measures from the dataset of Figure 5 have been done. In particular the Figure 6(a) shows the trajectory estimated using only the odometric data. In this case the error in the angle measurement is noticeably increased with respect to the one considered in previous examples and the trajectory estimated is far from the real one. In the second test the hypothesis of orthogonal features has been added: Figure 6(b) shows the results obtained. The robot maintains a good localization all along the path and the number of features in the map is further reduced. This test shows that the developed algorithm is able to correct the trajectory and gives a consistent map, confirming the high performance of the proposed algorithm and improving the quality of the map for the considered environments. Results in Figure 5 can be compared with results in Figure 6(b): in Figure 5 the same wall is generally represented by multiple features with slight different orientations; in Figure 6(b) instead, each wall is represented by only one feature. Note that open doors, represented by non-orthogonal short lines in Figure 5, disappear in Figure 6(b). The map resulting from such an algorithm is more compact, and well adapted to the explored office like environment. However, such a specialization could lead problems for a more general environment: in this case an algorithm integrating multiple primitives would have
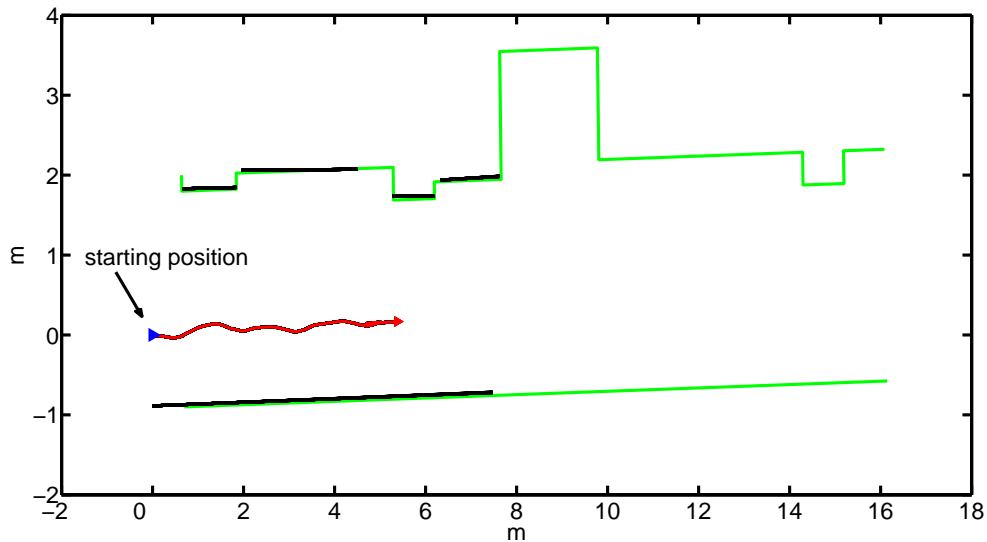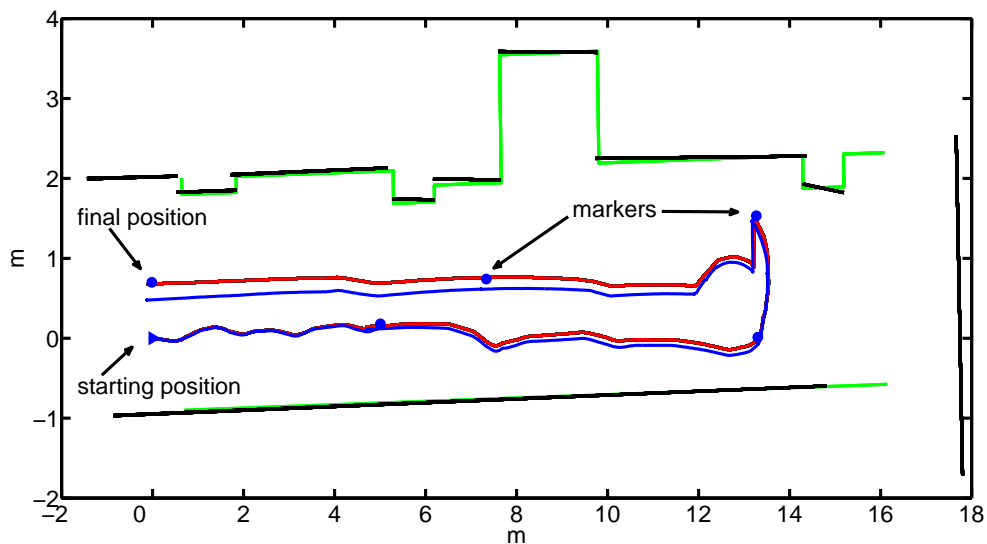
Fig. 3. (a) Estimated map and trajectory at $t = 80s$; (b) Estimated VS real map and odometric trajectory. The unit of coordinate-axis is meter for both panels.

better results, at loss of some computational efficiency.

6 Concluding remarks

In this paper, an implementation of the FastSLAM algorithm with a line features based map is developed. The line feature map is proven to be an effective solution in indoor semi structured environments: the obtained maps give a detailed and compact representation of the environment, and the map-
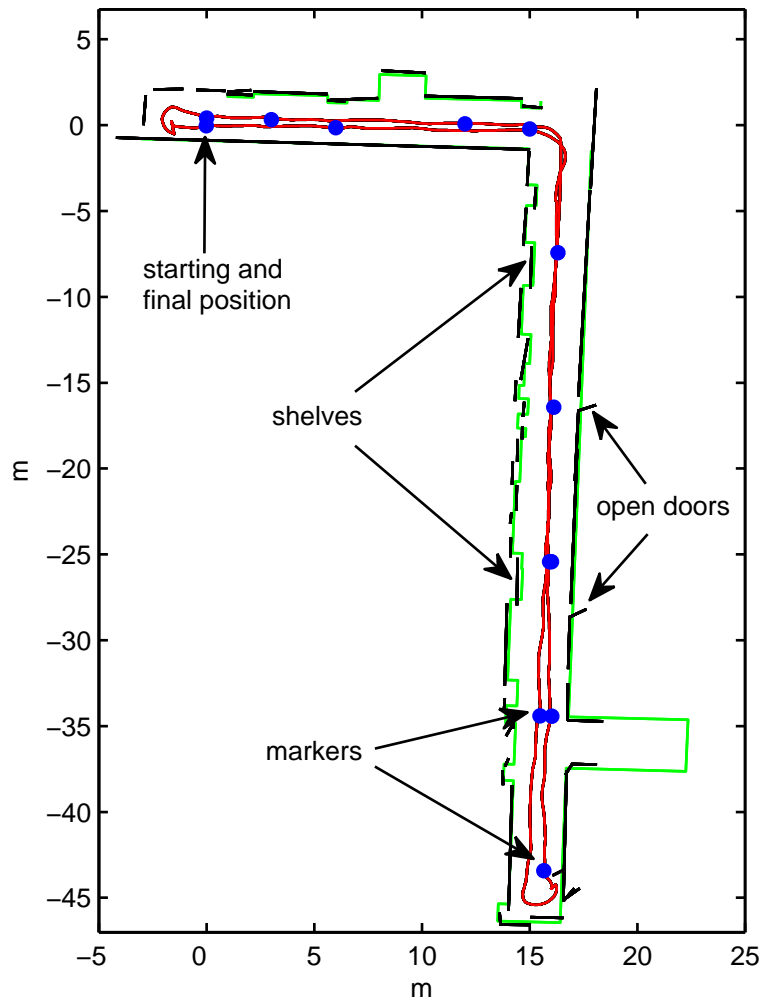
Fig. 4. Map of two Department corridors; 130m long trajectory. The unit of coordinate-axis is meter.

ping algorithm allows to estimate the measures referred to previously unseen parts of detected lines, reducing the localization and mapping uncertainty all along the path. The hypothesis of orthogonal features enhances these qualities in the considered environment. Thanks also to the gyroscope sensor accuracy, the number of particles needed to achieve good localization and mapping is reduced to few tens. The implemented solution has been tested with experiments in a completely unknown indoor environment. However, to use the algorithm in a more general environment, a more extended set of primitives would be necessary.

The analyzed strategy is a computationally efficient solution to the mobile base localization in an unknown office-like environment and it represents a suitable solution to develop a navigation system aimed at enhancing the efficiency and the security of smart wheelchairs; the obtained map is suitable for the vehicle

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

and the human user to interact in order to achieve a better knowledge of the environment and a safer navigation.

References

[1] R. Smith, P. Cheeseman, On the representation and estimation of partial uncertainty, Int. Journal of Robotics Research 5 (1986) 56–68.

[2] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, 2005.

[3] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping (SLAM): part i, IEEE Robotics & Automation Magazine 13 (2) (2006) 99–110.

[4] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): part ii, IEEE Robotics & Automation Magazine 13 (3) (2006) 108–117.

[5] A. Bicchi, F. Lorussi, P. Murrieri, V. Scordio, On the problem of simultaneous localization, map building, and servoing of autonomous vehicles, in: B. Siciliano, A. De Luca, C. Melchiorri, G. Casalino (Eds.), Advances in Control of Articulated and Mobile Robots, Vol. 10, Springer-Verlag, Berlin, Heidelberg, Germany, 2004, pp. 223–242.

[6] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem, IEEE Transactions on Robotics and Automation 17 (3) (2001) 229–241.

[7] A. Garulli, A. Giannitrapani, A. Rossi, A. Vicino, Mobile robot SLAM for line-based environment representation, in: Proceedings of the 44th IEEE Conference

Fig. 6. (a) Trajectory estimated by odometric data only; (b) Map and trajectory obtained by the FastSLAM algorithm with orthogonal features hypothesis and odometric and laser data only. The unit of coordinate-axis is meter for both panels.

on Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC '05), Seville, Spain, 2005, pp. 2041–2046.

[8] G. Ippoliti, L. Jetto, A. L. Manna, S. Longhi, 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, 2005, pp. 1463–1470.

[9] A. Bonci, G. Ippoliti, A. L. Manna, S. Longhi, L. Sartini, Sonar and video data fusion for robot localization and environment feature estimation, in: Proc. of the 44th IEEE Conference on Decision and Control, 2005 and 2005 European

Control Conference (CDC-ECC05), Seville, Spain, 2005, pp. 8337–8342.

[10] M. Montemerlo, S. Thrun, D. Koller, B. Webright, FastSLAM: a FActored Solution to the Simultaneous Localization And Mapping problem, in: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02), 2002, pp. 593–598.

[11] J. Nieto, J. Guivant, E. Nebot, Real Time Data Association for FastSLAM, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2003, pp. 412–418.

[12] D. Hahnel, W. Burgard, D. Fox, S. Thrun, An efficient FastSLAM Algorithm for generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurement, in: Proc. IEEE Int. Conf. on Intelligent Robots and Systems, 2003, pp. 206–211.

[13] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, D. Nardi, Fast and accurate SLAM with Rao-Blackwellized particle filters, Robot. Auton. Syst. 55 (1) (2007) 30–38.

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

[15] D. Ding, R. Cooper, Electric powered wheelchairs, IEEE Control Systems Magazine 25 (2) (2005) 22–34.

[16] T. Dutta, G. Fernie, Utilization of ultrasound sensors for anti-collision systems of powered wheelchairs, IEEE Transactions on Neural Systems and Rehabilitation Engineering 13 (1) (2005) 24–32.

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

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

[19] A. Doucet, On Sequential Simulation-Based Methods for Bayesian Filtering, Tech. Rep. CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering (1998).

[20] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo Localization for Mobile Robots, in: Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 1, 1999, pp. 1322–1328.

[21] K. Murphy, S. Russel, Rao-Blackwellised particle filtering for dynamic Bayesian networks, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), In Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001, pp. 176–183.

[22] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001.

[23] A. Bonci, G. Ippoliti, L. Jetto, T. Leo, S. Longhi, Methods and algorithms for sensor data fusion aimed at improving the autonomy of a mobile robot, in: B. Siciliano, A. De Luca , C. Melchiorri, G. Casalino (Eds.), Advances in Control of Articulated and Mobile Robots, Vol. 10, Springer-Verlag, Berlin, Heidelberg, Germany, 2004, pp. 191–222.

[24] G. Ippoliti, L. Jetto, S. Longhi, Localization of mobile robots: development and comparative evaluation of algorithms based on odometric and inertial sensors, Journal of Robotic Systems 22 (2005) 725–735.

[25] B. Barshan, H. Durrant-Whyte, Inertial navigation systems for mobile robots, IEEE Transactions on Robotics and Automation 11 (3) (1995) 328–342.

[26] K. Komoriya, E. Oyama, Position estimation of a mobile robot using optical fiber gyroscope (OFG), in: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', (IROS'94), Vol. 1, Munich, Germany, 1994, pp. 143–149.

[27] L. Ojeda, H. Chung, J. Borenstein, Precision-calibration of fiber-optics gyroscopes for mobile robot navigation, in: Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA, USA, 2000, pp. 2064–2069.

[28] C.M. Wang, Location estimation and uncertainty analysis for mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1988, pp. 1230–1235.

[29] C. Ye, J. Borenstein, Characterization of a 2-D laser scanner for mobile robot obstacle negotiation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02), Vol. 3, Washington, DC USA, 2002, pp. 2512 – 2518.

[30] G. Ippoliti, L.Jetto, S. Longhi, A. Monteri, A properly designed extended Kalman filtering approach for robot localization by sensors with different degree of accuracy, in: Proceedings of the fourth International Symposium on Robotics an Automation (ISRA 2004), Queretaro, Mexico, 2004, pp. 574–581.

[31] L. Zhang, B. Ghosh, Line Segment Based Map Building and Localization Using 2D Laser Rangefinder, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2000, pp. 2538–2543.

[32] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online Nonlinear/Non-Gaussian Bayesian tracking, IEEE, Transaction on Signal Processing 50 (2) (2002) 174–188.

[33] E. Brunskill, N. Roy, SLAM using Incremental Probabilistic PCA and Dimensionality Reduction, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2005, pp. 342 – 347.

[34] P. Newman, J. Leonard, J. Tardos, J. Neira, Explore and return: experimental validation of real-time concurrent mapping and localization, in: Proceedings of the Int. Conf. on Robotics and Automation, 2002, pp. 1802–1809.

[35] TGR, s.r.l, 40064 Ozzano dell' Emilia, Bologna, Italy.
URL `http://www.tgr.it`

[36] Sicod, s.r.l, 10093 Collegno (TO), Italy.
URL `http://www.sicod.it/`

[37] Hitachi cable, ltd.
URL `http://www.hitachi-cable.co.jp/en/index.html`

[38] SICK, s.p.a., 20090 Vimodrone (MI), Italy.
URL `http://www.sick.it/it/it.html`