An Asynchronous Multi-rate Approach to Probabilistic Self-Localisation and Mapping

L. Armesto† G. Ippoliti‡ S. Longhi‡ J. Tornero†

†DISA, Technical University of Valencia, Camino de Vera s/n 46022, Valencia, Spain

‡DIIGA, Universita Politecnica delle Marche, Via Brecce Bianche, 60131, Ancona, Italy

***Abstract*— In this paper, we present a set of robust and efficient algorithms with $\mathcal{O}(N)$ cost for the solution of the Simultaneous Localization And Mapping (SLAM) problem of a mobile robot. First, we introduce a novel object detection method, which is mainly based on multiple line fitting method for landmark detection with regular constrained angles. Second, a line-based pose estimation method is proposed, based on Least-Squares (LS). This method performs the matching of lines, providing the global pose estimation under assumption of known Data-Association. Finally, we extend the FastSLAM (FActored Solution To SLAM) algorithm for mobile robot self-localisation and mapping by considering the asynchronous sampling of sensors and actuators. In this sense, multi-rate asynchronous holds are used to interface signals with different sampling rates. Moreover, an asynchronous fusion method to predict and update mobile robot pose and map is also presented. In addition to this, FastSLAM 1.0 has been also improved by considering the estimated pose with the LS-approach to re-allocate each particle of the posterior distribution of the robot pose. This approach has a lower computational cost than the original Extended Kalman Filtering (EKF) approach in FastSLAM 2.0. All these methods have been combined in order to perform an efficient and robust self-localization and map building process. Additionally, these methods have been validated with experimental real data, in mobile robot moving on an unknown environment for solving the SLAM problem.**

***Index Terms*— Multi-rate fusion, probabilistic, localisation, mapping, FastSLAM.**

I. Introduction

One of the most common techniques for state estimation of non-linear discrete-time dynamic systems is the Extended Kalman Filter (EKF) [1] or more recently Unscented Kalman Filter (UKF) [2]. Kalman filter gives a robust, optimal and recursive state estimation to fuse redundant sensor information. However, both approaches assume that the probability distribution function (*pdf*) is Gaussian, which is not true for most signals found in practice. Other recent filtering methods are Particles Filters (PF) [3], [4], where the main advantage is that the *pdf* can be accurately approximated with a large number of particles. The most common approach to PF is the Sampling Importance Resampling (SIR) [5], [6] which provides a weight (importance factor) to each particle. The resampling step selects those particles with higher weights and removes those particles with lower weights. Another well-known approach is the Rao-Blackwellized PF [7] that uses a PF for some variables of the state and an Kalman filter for other variables.

In robotics, these estimation methods are commonly used to determine the robot pose with respect to its environment (map), see for instance [1], [8], [9], [10], [11], [12]. The problem of determining the robot pose is commonly known as *self-localisation*. The map can be assumed to be known or unknown. In the former the EKF, UKF and also the Monte Carlo Localization (MCL) methods [11], [12] have been widely used. In the latter, the SLAM (Simultaneous Localisation and Map Building) problem arises, which consists on estimating simultaneously the robot pose and map. Classical approaches to solve this problem are based on EKF by joining the robot state and the map [1], [13], [14]. The problem of this approach is that the dimensions of the covariance grows to $(N+3)\times(N+3)$, being N the number of map features.

Recently, a solution of the SLAM problem with Rao-Blackwellized PF has been given, known as FastSLAM (Factored Solution to SLAM) [15]. The key idea behind the FastSLAM is that the problem can be divided into $N+1$ separated problems. One for estimating the robot pose with a PF, and therefore with low state dimension; and N low dimension separate problems for updating the map features.

One of the main contributions of this paper is related to multi-rate asynchronous filtering approach for the SLAM problem based on PF. Previous multi-rate filter contributions are mainly for linear systems. In [16], [17], a Kalman filter is applied for LQG control, while in [18] a Kalman filter is developed using lifting techniques [19]. The problem of multi-rate filtering arises from the fact that sensors and actuators of robots are sampled at different sampling rates due to technological limitations, communication channels, processing time, etc.

In the paper, significant improvements for robot pose estimation are obtained when introducing multi-rate techniques to FastSLAM. In particular, it is shown that multi-rate fusion aims to provide more accurate results in loop closing problems in SLAM (localization and map building problems with closed paths).

Additionally, in this paper a pose estimation algorithm based on Least-Squares (LS) fitting of line features is proposed. Since the complexity of LS fitting is linear to the number of features, this implies a low computational cost than other techniques. Therefore, methods based on PF such as MCL and FastSLAM which requires a large number of particles may get benefit from this fact. In particular, this provides an accurate approximation of the posterior probability distribution function *pdf* for FastSLAM 2.0 [20].

Moreover, we developed an asynchronous filtering method to deal with measurements of sensors at different sampling rates. One of the key ideas of the method is to use an asynchronous hold to extrapolate inputs of the system. Another key point is the asynchronous execution of prediction and update steps in the filtering method, which aims to maintain a good system performance. The prediction step is executed within at least a pre-specified sampling period (generally at a fast sampling rate to reduce discretization errors) and update step is executed only when measurements are asynchronously received.

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. These applications are of interest in the emerging area of assistance technologies where powered wheelchairs can be used to strengthen the residual abilities of users with motor disabilities [21], [22], [23].

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.

This paper is organised as follows: section II describes a method for multiple line fitting, pre-processing raw laser measurements; section III describes the LS pose estimation method; section IV shows probabilistic approaches to the SLAM problem as well as the proposed multi-rate asynchronous approach to FastSLAM. Some experimental results can be found in section V and the conclusions in section VI.

II. Object Detection

In this section, a method for object detection is described, based on laser ranger measurements. This method is specifically recommended for indoor applications, since it assumes certain geometric constraints about the environment.

Firstly, we define an object as a set of points $\{x_i, y_i\}$, representing a given landmark of the environment (walls, shelves, columns, corners, persons, etc.). In particular, for localization purposes, the map contains lines representing walls, shelves, doors, etc. Moving objects such as persons are not included in the map, since it is assumed to be static.

A. Segmentation and Single Line Fitting

Laser points are segmented using an adaptive breakpoint detector [24] and the well-know Split and Merge algorithm [25]. Initially, a standard LS fitting is used for each segmented line to estimate distance (ρ) and angle (φ) parameters, where the well-known solution is:

$$\hat{\rho} = \bar{x}\cos\hat{\varphi} + \bar{y}\sin\hat{\varphi},\ \ \hat{\varphi} = \frac{1}{2}\arctan\ \ \frac{-2\sigma_{xy}^2}{\sigma_y^2 - \sigma_x^2}\ \ ,$$

where $\bar{x}$, $\bar{y}$, σ_x^2, σ_y^2 and σ_{xy}^2 are means and covariances of the data set.

B. Multiple Line Fitting

Since our object detection algorithm is designed for structured indoor environments, it is reasonable to make some assumptions about the properties of the environment. In particular, walls are usually perpendicular or parallel to each other. With that in mind, we present a method for fitting lines by considering such as geometric constraints of the environment.

The goal of this method is the minimization of the sum of squared distances for all lines (global multiple line fitting). This basic idea allows us to improve typically noisy estimations of lines with few points. In general, the method may not only improve parameter estimation but also data-association and map-building. In addition to this, corners can easily be estimated as intersection points of contiguous lines.

By performing a previous single estimation (standard LS fitting), we can select those lines whose angle difference is closed to $\varphi_k - \varphi_0 \approx \{0^{\rm o}, 90^{\rm o}, 180^{\rm o}, 270^{\rm o}\}$ with respect to a reference line. The idea is to force the estimation to generate $\varphi_k - \varphi_0 = \{0^{\rm o}, 90^{\rm o}, 180^{\rm o}, 270^{\rm o}\}$. The reference line should be taken to be the most reliable one, i.e. the largest line or the line with the greatest number of points. Obviously, the method is not limited to rectangular angles and any angle might be forced. If a line does not satisfy the constrained condition, it is not forced and the single line fitting estimation is used instead.

The performance index to minimize is as follows:

$$E_{ML} = \frac{1}{2}\sum_{k=1}^{N_l}\sum_{i=1}^{N_{l,k}}\left(\rho_k - x_{i,k}\cos\varphi_k - y_{i,k}\sin\varphi_k\right)^2,$$

constrained to $\varphi_k = \varphi_0 + \varphi_{c,k}$, where $\varphi_{c,k}$ is the constrained angle with respect to the reference line (for convenience the first line); N_l is the number of lines and $N_{l,k}$ the number of points for each segmented line.

The distances of lines can be obtained as follows:

$$\frac{\partial E_{ML}}{\partial \rho_k} = 0 \Rightarrow\ \ \hat{\rho}_k = \bar{x}_k\cos\hat{\varphi}_k + \bar{y}_k\sin\hat{\varphi}_k,$$

Replacing these variables and angle constrains into the original performance index and performing some standard algebraic manipulations, we get to the optimal estimate of φ_0:

$$\frac{\partial E_{ML}}{\partial \varphi_0} = \tilde{B}\sin(2\hat{\varphi}_0) - \tilde{A}\cos(2\hat{\varphi}_0) = 0 \Rightarrow \tan(2\hat{\varphi}_0) = \frac{\tilde{A}}{\tilde{B}},$$

with

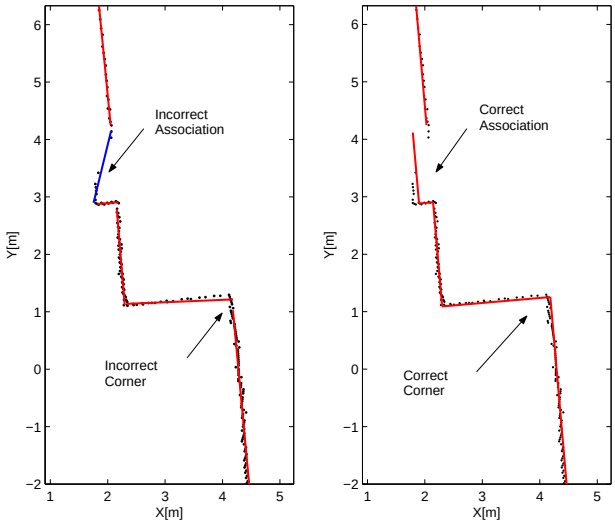
$$\tilde{A} = -\sum_{k=0}^{N_{l,k}} 2\sigma_{xy,k}^2 \cos(2\varphi_{c,k}) + (\sigma_{y,k}^2 - \sigma_{x,k}^2)\sin(2\varphi_{c,k}),$$
$$\tilde{B} = \sum_{k=0}^{N_{l,k}} -2\sigma_{xy,k}^2 \sin(2\varphi_{c,k}) + (\sigma_{y,k}^2 - \sigma_{x,k}^2)\cos(2\varphi_{c,k}).$$

It is interesting to remark that data statistics, that is $\bar{x}$, $\bar{y}$, σ_x^2, σ_{xy}^2 and σ_y^2, were initially computed for single fitting and therefore they are reused for saving computational resources. Algorithm 1 describes the implementation of the proposed method. The input of the algorithm are polar raw data of the laser ranger $(\mathbf{r}_t, \boldsymbol{\alpha}_t)$, while the output of the algorithm is the vector of detected lines $\mathbf{z}_t$.

Algorithm 1: Multiple line fitting

MultipleLineFitting($\mathbf{r}_t$, $\boldsymbol{\alpha}_t$)
remove those measurements from $\mathbf{r}_t$ and $\boldsymbol{\alpha}_t$ higher than a maximum distance;
compute break points based on discontinuities using $(\mathbf{r}_t, \boldsymbol{\alpha}_t)$;
compute Cartesian points $(\mathbf{x}_t, \mathbf{y}_t)$ from $(\mathbf{r}_t, \boldsymbol{\alpha}_t)$;
segment Cartesian points with Split & Merge algorithm;
estimate lines $\mathbf{z}_t$ with LS fitting, one for each segmented set of points (single fitting);
retrieve $\hat{\varphi}_0$ from the largest line of $\mathbf{z}_t$;
$\tilde{A} = 0$; $\tilde{B} = 0$;
for $i = 1$ **to** $\text{length}(\mathbf{z}_t)$ **do**
  retrieve $\langle \hat{\varphi}_i, \sigma_{x,i}^2, \sigma_{xy,i}^2, \sigma_{y,i}^2 \rangle$ from $\mathbf{z}_t^i$;
  if $\hat{\varphi}_i - \hat{\varphi}_0 \simeq 0$ **then** $\hat{\varphi}_{c,i} = \hat{\varphi}_0$;
  else if $\hat{\varphi}_i - \hat{\varphi}_0 \simeq \frac{\pi}{2}$ **then** $\hat{\varphi}_{c,i} = \hat{\varphi}_0 + \frac{\pi}{2}$;
  else if $\hat{\varphi}_i - \hat{\varphi}_0 \simeq \pi$ **then** $\hat{\varphi}_{c,i} = \hat{\varphi}_0 + \pi$;
  else if $\hat{\varphi}_i - \hat{\varphi}_0 \simeq \frac{3\pi}{2}$ **then** $\hat{\varphi}_{c,i} = \hat{\varphi}_0 + \frac{3\pi}{2}$;
  else $\hat{\varphi}_{c,i} = \hat{\varphi}_i$;
  $\tilde{A} = \tilde{A} - 2\sigma_{xy,i}^2 \cos(2\hat{\varphi}_{c,i}) - (\sigma_{y,i}^2 - \sigma_{x,i}^2)\sin(2\hat{\varphi}_{c,i})$;
  $\tilde{B} = \tilde{B} + (\sigma_{y,i}^2 - \sigma_{x,i}^2)\cos(2\hat{\varphi}_{c,i}) - 2\sigma_{xy,i}^2 \sin(2\hat{\varphi}_{c,i})$;
end
$\hat{\varphi}_{c,0} = \frac{1}{2}\arctan(\frac{\tilde{A}}{\tilde{B}})$;
for $i = 1$ **to** $\text{length}(\mathbf{z}_t)$ **do**
  retrieve $\langle \hat{\rho}_i, \bar{x}_i, \bar{y}_i \rangle$ from $\mathbf{z}_t^i$;
  $\hat{\varphi}_i = \hat{\varphi}_{c,i} + \hat{\varphi}_{c,0}$;
  $\hat{\rho}_i = \bar{x}\cos\hat{\varphi}_i + \bar{y}\sin\hat{\varphi}_i$;
  replace $\langle \hat{\rho}_i \ \hat{\varphi}_i \rangle$ of $\mathbf{z}_t^i$;
end
return $\mathbf{z}_t$;

A comparison between the single and multiple line fitting methods is shown in Figure 1. It can be appreciated that the multiple line fitting method gives better estimation in the sense that lines are correctly estimated and corners are better defined. Therefore, this will improve the global estimation and map building, without additional efforts, since the cost of this multiple line fitting is always of order $\mathcal{O}(N_l)$ as in the single-line fitting.



(a) Single line fit (b) Multiple line fit

Fig. 1. Comparative between single and multiple line fit.

III. Least-Squares Pose Estimation

In this section, we present a simple but effective method to estimate the pose of a robot based on line features. The method assumes that a data pre-process has been previously done, so detected line features of the environment have been obtained $\mathbf{z}_t^i = \langle \rho_d^i, \varphi_d^i \rangle$. It also assumes that the map, containing line features $\mathbf{m}_t^j = \langle \rho_m^j, \varphi_m^j \rangle$, is known or it is being estimated.

The main issue of this method is that it estimates the robot pose in LS sense and it can be applied for global localization under the assumption of known data association. Since this is not a realistic approach in most real applications, the method requires a previous association between detected and map features, which is given by the hypothesis $\mathcal{H}$, a list that relates each detected i feature with its corresponding feature on the map $j = \mathcal{H}(i)$. If a feature is not associated, then we set $\mathcal{H}(i) = 0$.

Based on the well-known line model:

$$\rho_d^i = \rho_m^{\mathcal{H}(i)} - x\cos\varphi_m^{\mathcal{H}(i)} - y\sin\varphi_m^{\mathcal{H}(i)} + \epsilon_{\rho,i}, \quad (1)$$
$$\varphi_d^i = \varphi_m^{\mathcal{H}(i)} - \theta + \epsilon_{\varphi,i}, \quad (2)$$

where $\mathbf{p} = [x \ y]^T$ is the Cartesian robot position and θ the orientation; $\epsilon_{\rho,i}$ and $\epsilon_{\varphi,i}$ are distance and angle errors between detected lines and predicted ones. Error covariances are assumed to be known, $\Sigma_\rho = \sigma_\rho^2 \mathbf{I}$ and $\Sigma_\varphi = \sigma_\varphi^2 \mathbf{I}$, respectively.

The pose estimation problem can be easily separated because equation (1) is only affected by the Cartesian pose, while equation (2) is only affected by the orientation. In addition to this, we can easily see that both equations are linear with respect to the pose variables and they can be rewritten in matrix form as:

$$\boldsymbol{\rho}_e = -\mathbf{X} \cdot \mathbf{p} + \boldsymbol{\epsilon}_\rho,$$
$$\boldsymbol{\varphi}_e = [1 \ \ldots \ 1]^T \theta + \boldsymbol{\epsilon}_\varphi,$$

where

$$\boldsymbol{\rho}_e = \left[\rho_d^1 - \rho_m^{\mathcal{H}(1)}, \ldots, \rho_d^{N_l} - \rho_m^{\mathcal{H}(N_l)}\right]^T$$

$$\mathbf{X} = \begin{bmatrix} \cos\varphi_m^{\mathcal{H}(1)} & \sin\varphi_m^{\mathcal{H}(1)} \\ \vdots & \vdots \\ \cos\varphi_m^{\mathcal{H}(N_l)} & \sin\varphi_m^{\mathcal{H}(N_l)} \end{bmatrix},$$

$$\boldsymbol{\varphi}_e = \left[\varphi_d^1 - \varphi_m^{\mathcal{H}(1)}, \ldots, \varphi_d^{N_l} - \varphi_m^{\mathcal{H}(N_l)}\right]^T,$$

$$\boldsymbol{\epsilon}_\rho = [\epsilon_{\rho,1}, \ldots, \epsilon_{\rho,N_l}]^T,\ \boldsymbol{\epsilon}_\varphi = [\epsilon_{\varphi,1}, \ldots, \epsilon_{\varphi,N_l}]^T.$$

Therefore, the LS fitting provides the following estimations for the cartesian pose and orientation

$$\hat{\mathbf{p}} = \begin{matrix} \hat{x} \\ \hat{y} \end{matrix} = -(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\rho}_e,\ \hat{\theta} = -\bar{\boldsymbol{\varphi}}_e,$$

and the covariance estimation error is as follows:

$$\Sigma_{xy} = \sigma_\rho^2(\mathbf{X}^T\mathbf{X})^{-1},\ \Sigma_\theta = \frac{\sigma_\varphi^2}{N_l}.$$

Matrix $\mathbf{X}^T\mathbf{X}$ is singular when all lines considered have the same orientation. However, because of the method estimates the Cartesian position separately from the orientation, in such as situations a valid estimation of the orientation can still be provided.

Algorithm 2 describes the implementation of the method, where the following considerations must be taken into account:

$$\mathbf{X}^T\mathbf{X} = \begin{matrix} A & C \\ C & B \end{matrix},\ \mathbf{X}^T\boldsymbol{\rho}_e = \begin{matrix} D \\ E \end{matrix},$$

with

$$A = \sum_{i=1}^{N_l}\cos^2\varphi_m^{H(i)},\ B = \sum_{i=1}^{N_l}\sin^2\varphi_m^{\mathcal{H}(i)},\ C = \sum_{i=1}^{N_l}\cos\varphi_m^{\mathcal{H}(i)}\sin\varphi_m^{\mathcal{H}(i)},$$

$$D = \sum_{i=1}^{N_l}(\rho_d^i - \rho_m^{\mathcal{H}(i)})\cos\varphi_m^{\mathcal{H}(i)},\ E = \sum_{i=1}^{N_l}(\rho_d^i - \rho_m^{\mathcal{H}(i)})\sin\varphi_m^{\mathcal{H}(i)}.$$

A similar approach is followed by Araujo and Aldon in [26], where the main difference is given on the definition of the performance index. In [26], the proposed index is based on the minimization of Cartesian projections between detected lines and map lines with respect to robot pose. This representation introduces the advantage of including points and lines under the same performance index, but the analytic solution to this problem leads to solve a 4^{th} order polynomial equation derived from a trigonometric rotation equation. In addition, it presents 4 possible solutions (real and/or complex) and therefore, a study on the Jacobian is required to discard local maxima.

In order to analyze the performance of the proposed method, a simulation, using a mobile robot moving through a regular environment with walls and corridors, has been considered. Obviously, the line detection method proposed in Section II has been taken into account for laser-ranger measurements, where a noise has been added to them to simulate more realistic data. Figure 2 shows the pose estimation error between *ground-truth* and estimated pose with LS and Araujo's methods. Crosses depicted on Figures 2(a) and 2(b) represent iterations where $\mathbf{X}^T\mathbf{X}$ becomes singular and therefore they should not be considered. It can be appreciated that Araujo's method presents more than three times higher cumulative computational time (T_c) than LS Pose Estimation. In fact, the mean computational time of LS and Araujo's methods are $4.92 \cdot 10^{-4}$ms and $18 \cdot 10^{-4}$ms, respectively. This aspect is particularly relevant, since each particle of the FastSLAM algorithm will correct its pose using this method (see section IV for more details).

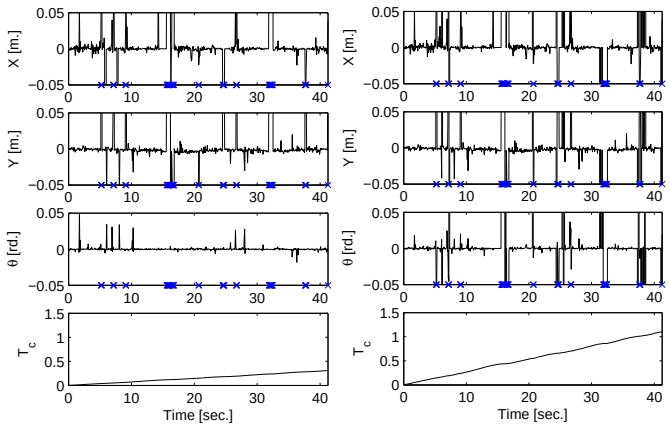
Moreover, consider the Mean Estimation Error (MSE) of Cartesian positions and orientations:

$$MSE_{xy} = \frac{1}{h}\sum_{k=0}^{h}(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2,\ MSE_\theta = \frac{1}{h}\sum_{k=0}^{h}(\theta_k - \hat{\theta}_k)^2,$$

where h is the number of iterations of the simulation. The performance in terms of MSE_x and MSE_θ of LS and Araujo's methods is shown in Table I, where N_l is the minimum number of lines required to produce a valid estimation.

Algorithm 2: Line-Based Pose Estimation.

PoseEst($\mathbf{z}_t$, $\mathbf{m}_t$, $\mathcal{H}_t$, $\mathbf{x}_{t-1}$)
$A = B = C = D = E = \varphi_e = N_l = 0$;
for $i = 0$ **to** $\text{length}(\mathcal{H}_t)$ **do**
  if $\mathcal{H}_t(i) > 0$ **then**
    retrieve $\langle \rho_m^j, \varphi_m^j \rangle$ from $\mathbf{m}_t^{\mathcal{H}_t(i)}$;
    retrieve $\langle \rho_d^i, \varphi_d^i \rangle$ from $\mathbf{z}_t^i$;
    $A = A + \cos^2\varphi_m^j$;
    $B = B + \sin^2\varphi_m^j$;
    $C = C + \sin\varphi_m^j\cos\varphi_m^j$;
    $D = D + (\rho_d^i - \rho_m^j)\cos\varphi_m^j$;
    $E = E + (\rho_d^i - \rho_m^j)\sin\varphi_m^j$;
    $\varphi_e = \varphi_e + \varphi_d^i - \varphi_m^j$;
    $N_l = N_l + 1$;
  end
end
retrieve $\langle x_t, y_t, \theta_t \rangle$ from $\mathbf{x}_{t-1}$;
if $j > 0$ and $A \cdot B - C^2 \neq 0$ **then**
  $x_t = \frac{C \cdot E - B \cdot D}{A \cdot B - C^2}$;
  $y_t = \frac{C \cdot D - A \cdot E}{A \cdot B - C^2}$;
  $\Sigma_{xy} = \sigma_\rho^2 \begin{bmatrix} A & C \\ C & B \end{bmatrix}^{-1}$;
else $x_t = x_{t-1}$; $y_t = y_{t-1}$; $\Sigma_{xy} = \mathbf{0}$;
if $j > 0$ **then** $\theta_t = -\frac{\varphi_e}{N_l}$; $\Sigma_\theta = \frac{\sigma_\varphi^2}{N_l}$;
else $\theta_t = \theta_{t-1}$; $\Sigma_\theta = 0$;
$\mathbf{x}_t = [x_t\ y_t\ \theta_t]^T$;
$\Sigma_{\mathbf{x}_t} = \begin{bmatrix} \Sigma_{xy} & \mathbf{0} \\ \mathbf{0}^T & \Sigma_\theta \end{bmatrix}$;
return $\mathbf{x}_t$ and $\Sigma_{\mathbf{x}_t}$;



(a) LS Pose Estimation (b) Araujo's method

Fig. 2. Pose estimation error and computational time with LS Pose Estimation and Araujo's methods.

MSE	Method	$N_l \geq 2$	$N_l \geq 3$	$N_l \geq 4$
$MSE_x[m^2]$	LS	$1.8{\cdot}10^{-3}$	$2{\cdot}10^{-3}$	$2.2{\cdot}10^{-3}$
	Araujo	27.85	$7.42\cdot10^{-5}$	$6.26{\cdot}10^{-5}$
$MSE_\theta[rd^2]$	LS	$1.75{\cdot}10^{-5}$	$1.93{\cdot}10^{-5}$	$1.53{\cdot}10^{-5}$
	Araujo	0.32	$7.7\cdot10^{-6}$	$4.74{\cdot}10^{-6}$

TABLE I

MSE PERFORMANCE.

Araujo's method gives more accurate results than LS on the cases where more than two lines have been detected ($N_l \geq 3$ or $N_l \geq 4$). However, it is clear that the accuracy of both methods is good enough for many practical situations.

IV. Asynchronous Multi-rate Techniques Applied To SLAM Problem

A. Probabilistic Robot Localization and Map Building

The key idea of particle filters (PF) is to represent the posterior distribution of a signal $\mathbf{x}_t$ by a set of random samples (particles) drawn from this posterior. Such a set of samples is an approximation of the "real" distribution and, in general, PF can represent a much more broader spaces of distributions, rather than Gaussian distributions like EKF or UKF.

We denote the set of particles, describing the posterior distribution as,

$$\mathcal{X}_t = \{\mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \ldots, \mathbf{x}_t^{[M]}\},$$

where M is the number of particles and each particle $\mathbf{x}_t^{[k]}$ is drawn from the posterior distribution:

$$\mathbf{x}_t^{[k]} \sim p(\mathbf{x}_t | \, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}), \tag{3}$$

where $\mathbf{z}_{1:t}$ represents the set of whole measurements and $\mathbf{u}_{1:t}$ the set of whole inputs. The approximation is valid if the number of samples is large enough, generally $M \geq 100$. Like most popular filters, the PF uses a recursive estimation for approximating the posterior distribution. Therefore, equation (3) simplifies to,

$$\mathbf{x}_t^{[k]} \sim p(\mathbf{x}_t | \, \mathbf{x}_{t-1}, \mathbf{z}_t, \mathbf{u}_t),$$

where only inputs and measurements at the present time instant are used. The main problem is that the posterior distribution is not known and therefore a proposal distribution has to be used, based on the *importance factor*. For each particle, a weight obtained from $w_t^{[k]} = p(\mathbf{z}_t|\mathbf{x}_t^{[k]})$ is computed, where $p(\mathbf{z}_t|\mathbf{x}_t)$ is the *pdf* of the sensor model. The set of weighted particles represents (an approximation) of the posterior. This method is known as **Sequential Importance Resampling** (SIR), where a resampling step and normalization must be done in order to avoid the algorithm degeneracy [6], [27].

FastSLAM is the PF approach to SLAM. Unfortunately PF in SLAM are also affected by the *curse of dimensionality*, with exponential grow ($M \times N$). For that reason, Rao-Blackwellized particle filters [28] are applied, where the pose of the robot is estimated with a standard PF and the map is estimated by an EKF. In FastSLAM each particle contains an estimated robot pose $\mathbf{x}_t^{[k]}$ and a set of Kalman filters with $\boldsymbol{\mu}_t^{[k]}$ and covariance $\boldsymbol{\Sigma}_t^{[k]}$, one for each feature in the map. Thus the problem is decomposed in $N+1$ problem: 1 for estimating the robot pose and N for estimating the features in the map. A key advantage of this solution is that maps between particles are not correlated and therefore errors in the map are filtered through the resampling process, where only the best particles (with best maps) will survive.

The FastSLAM described so far is known as FastSLAM 1.0 [15], while in FastSLAM 2.0 [20] the proposal distribution takes the measurements $\mathbf{z}_t$ into account when sampling the pose $\mathbf{x}_t^{[k]}$ based on a EKF approach. This modification follows a similar approach than [29], [30] for PF.

In mobile robots, inputs and outputs have different sampling rates. For instance, odometry sensors, such as encoders, gyros, accelerometers, are typical sampled faster than external sensors such as lasers [31], sonars or vision systems [32], [33]. This is a problem that arises form the inherent technological limitations of sensors, communication channels, processing cost, etc., which can not be neglected. A typical solution to overcome this problem is to reduce the overall sampling period to the slowest one. However, it is well known that this approach may decrease the overall system performance. In addition to this, discretization errors will also increase with higher sampling periods, therefore faster dynamics require fast sampling rates.

In this paper, we describe an asynchronous filtering method to deal with measurements of sensors at different sampling rates. One of the key ideas of the method is to use an asynchronous hold to extrapolate the inputs of the system. Another key point is the asynchronous execution of prediction and update steps in the filtering method, which maintains a good system performance. The prediction step is executed within at least a pre-specified sampling period (generally at a fast sampling rate to reduce discretization errors) and update step is executed only when measurements are asynchronously received.

B. Asynchronous Holds

A multi-rate hold is a hybrid *device* for generating, from a sequence of inputs sampled at a slow sampling rate, a continuous signal which may be discretized at a high sampling rate. The mathematical background of multi-rate high-order holds (MR-HOH) and samplers is described in [34], [35], [36]. Initially, multi-rate holds were introduced as a generalization of classical holds such as zero, first and second order holds (ZOH, FOH and SOH). Later on, in [37], a wide variety of holds were obtained from general primitive functions. The idea behind general primitive functions is to generate an extrapolated continuous signal based on input samples $\{\mathbf{u}_{t_j}, \mathbf{u}_{t_{j-1}}, \ldots, \mathbf{u}_{t_{j-n}}\}$, uniformly distributed at a low frequency (dual-rate sampling) [37].

In this paper, we extend the formulation for the asynchronous case, where input samplings are not uniformly distributed:

$$\hat{\mathbf{u}}_t = \sum_{l=0}^{n} \mathbf{f}_l(t, \mathbf{u}_{t_{j-l}}, t_{j-l}), \tag{4}$$

where $\mathbf{u}_{t_j}$ denotes an input that has been sampled at time instant t_j, being $t_{j-n} \leq \ldots \leq t_j \leq t$. The primitive function $\mathbf{f}_l(t)$ generates the continuous signal $\hat{\mathbf{u}}_t$, which can be computed at any desired time instant. Thus, the asynchronous hold is in charge of generating a "continuized" signal regardless when the inputs were sampled. Asynchronous holds may be used in two different situations: for estimating signals in between samples as well as for overcoming the data-missing problem.

Algorithm 3 implements the asynchronous hold of order n, based on a general primitive function. In order to implement the asynchronous hold, a shift register $\mathcal{U}_j = \{\mathbf{u}_{t_j}, \ldots, \mathbf{u}_{t_{j-n}}\}$ is required to log the signal. Table II summarizes some primitive functions that can be used in asynchronous holds. In Spline Holds, a set of coefficients $c_{j,l}$ are obtained when a spline curve is adjusted to the previous inputs. In Taylor Holds, input derivatives are obtained using the backward approximation.

Algorithm 3: Asynchronous-Hold.

1 **Asynchronous-Hold**($\mathbf{u}_t$, $\mathcal{U}_j$, t, j)
2 **if** $\mathbf{u}_t$ is updated **then**
3 shift out $\mathbf{u}_{t_{j-n}}$ and shift in $\mathbf{u}_t$ from $\mathcal{U}_j$;
4 $\hat{\mathbf{u}}_t = \mathbf{u}_t$;
5 $j = t$;
6 **else**
7 $\hat{\mathbf{u}}_t = \mathbf{0}$;
8 **for** $l = 0$ **to** n **do**
9 retrieve $\mathbf{u}_{t_{j-l}}$ from $\mathcal{U}_j$;
10 $\hat{\mathbf{u}}_t = \hat{\mathbf{u}}_t + \mathbf{f}_l(t, \mathbf{u}_{t_{j-l}}, t_{j-l})$;
11 **end**
12 **end**
13 **return** $\hat{\mathbf{u}}_t$, $\mathcal{U}_j$ and j;

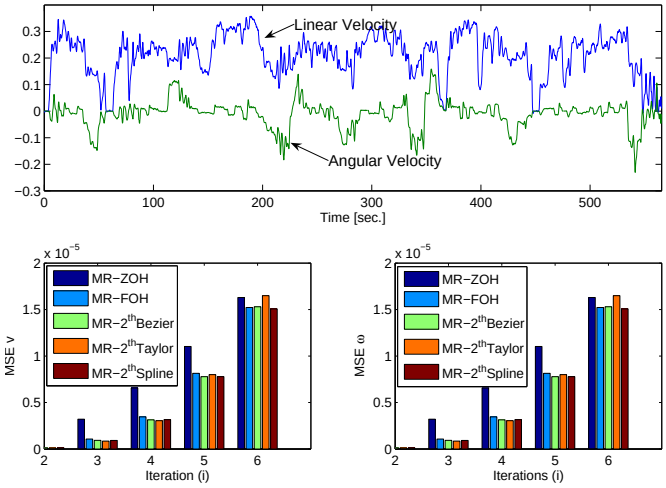


Fig. 3. Input profile and MSE for different Holds.

An experiment has been performed on a mobile robot moving on a real environment, where signal profiles of linear and angular velocities are shown in Figure 3. This figure also shows the MSE of the extrapolated signals when using a MR-FOH (1^{st} order "Lagrange" Hold), MR-SOBH (2^{nd} order Bezier hold), MR-SOTH (2^{nd} order Taylor hold) and MR-SOSH (2^{nd} order Spline Hold)[1]. The sampling period of signals is $T_s = 100ms$, i is the number of iterations that signals are extrapolated and n is the order of the hold. Obviously, stability can not be guaranteed for any arbitrary extrapolation time interval, since MSE increases exponentially with i. However, it can be seen that for all cases with $i \leq 5$ the results of MR-FOH, MR-SOBH, MR-SOTH and MR-SOSH significantly improve the results of the naive MR-ZOH solution.

C. Multi-rate Asynchronous FastSLAM

The proposed multi-rate asynchronous FastSLAM (MR-FastSLAM) has been implemented in the two different versions, FastSLAM 1.0 and FastSLAM 2.0, described in Algorithm 4, where M is the number of particles. In FastSLAM 1.0, pseudo-code lines 11 and 12 are not used. The pose estimation method used in line 2 is the one described in section III. This modifies the original contribution of FastSLAM [20], in the sense that a LS approach is used instead of an EKF approach. The main advantage of this new approach is the reduced computational cost of LS.

The steps are similar to the conventional FastSLAM algorithms, where differences lie on the inclusion of asynchronous holds and asynchronous prediction and update steps. In that sense, asynchronism is due to a time-varying execution time period t:

[1] 1^{st} order cases of Spline, Bezier and Taylor functions are the same that the FOH.

Type	Function
Interpolation (Lagrange)	$\hat{\mathbf{u}}_t = \sum_{l=0}^{n} \left[\prod_{\substack{q=0 \\ q \neq l}}^{n} \frac{t - t_{j-q}}{t_{j-l} - t_{j-q}} \quad \mathbf{u}_{t_{j-l}} \right]$
Interpolation (Spline)	$\hat{\mathbf{u}}_t = \sum_{l=0}^{n} c_{j,l}(t - t_{j-l})^l$ with $\{c_{j-n+1,0}, \ldots, c_{j,0}, \ldots, c_{j-n+1,m}, \ldots, c_{j,m}\} = \text{spline}(\mathcal{U}_{t_j})$
Approximation (Bezier)	$\hat{\mathbf{u}}_t = \sum_{l=0}^{n} \left[\frac{n!}{l!(n-l)!} \quad 1 + \frac{t - t_j}{t_j - t_{j-n}}^{n-l} \quad \frac{t - t_j}{t_j - t_{j-n}}^{l} \quad \mathbf{u}_{t_{j-l}} \right]$
Approximation (Taylor)	$\hat{\mathbf{u}}_t = \sum_{l=0}^{n} \frac{(t - t_j)^l}{l!} \mathbf{u}_{t_j}^{l)}$ with $\mathbf{u}_{t_j}^{l)} = \frac{\mathbf{u}_{t_j}^{l-1)} - \mathbf{u}_{t_{j-1}}^{l-1)}}{t_j - t_{j-1}}$

TABLE II

PRIMITIVE FUNCTIONS.

- If the time T_s has elapsed since the last execution $(t-1)$ period of the algorithm and no measurement has been received yet, then only the prediction step is executed with $t = (t-1) + T_s$ to maintain a good system performance with low discretization error.
- Whenever a measurement is received, the prediction and update steps are executed with $t = (t-1) + t_z$, with t_z being the elapsed time between the last execution and the measurement sampling. Note that the prediction step must necessarily be performed in order to coherently fuse measurements and states at the same time instant.

Figure 4 shows an example of tasks chronogram for the proposed algorithm. In this case, four different tasks have been considered: Encoder, Laser, Prediction and Update; each of them running at different frequencies. The Encoder task is executed approximately at $T_{inc} \approx 100ms$, each time an encoder measurement is received, while Laser Task is executed at $T_{las} \approx 400ms$. Laser Task has been sub-divided into more sub-tasks to indicate when each of detected lines have been processed, where it can be appreciated in Figure 4 that several lines are typically processed for a specific laser scan. Prediction and Update Tasks are executed according to conditions mentioned above. In this sense, the update step is not performed until the processing of laser scan has been completed. In Figure 4, it can also be seen that the Prediction Task is regularly executed with a sampling period of $T_s = 100ms$. However, whenever a laser measurement is received, Prediction Task is executed again, immediately followed by Update Task. In addition to this, it is interesting to observe the effect of the asynchronous hold: despite of asynchronous encoder samplings, the prediction step, which uses these measurements, is executed at its own sampling frequency (also asynchronous).

Algorithm 4: Multi-rate FastSLAM 2.0.

MR-FastSLAM 2.0($\mathcal{X}_{t-1}$, $\mathbf{u}_t$, $\mathbf{z}_t$, $\mathcal{U}_j$, t, j)
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \mathbf{z}_t^s = \emptyset$;
$[\hat{\mathbf{u}}_t, \mathcal{U}_j, j] = \text{Asynchronous-Hold}(\mathbf{u}_t, \mathcal{U}_j, t, j)$;
$[\hat{\mathbf{u}}_t, \mathcal{U}_t] = \text{MR-Hold}(\mathbf{u}_t, \mathcal{U}_t, t)$;
for $k = 1$ **to** M **do**
    retrieve $\langle \mathbf{x}_{t-1}^{[k]}, \mathbf{m}_{t-1}^{[k]}, w_{t-1}^{[k]} \rangle$ from $\mathcal{X}_{t-1}$;
    $\mathbf{x}_t^{[k]} = \text{MotionModel}(\hat{\mathbf{u}}_t, \mathbf{x}_{t-1}^{[k]}, t)$;
    if $\mathbf{z}_t$ has been sampled **then**
        $\hat{\mathbf{z}}_t = \text{MeasurementModel}(\mathbf{m}_{t-1}, \mathbf{x}_t^{[k]})$;
        $[\mathcal{H}_t, w_t^{[k]}] = \text{DataAssociation}(\mathbf{z}_t, \hat{\mathbf{z}}_t)$;
        $[\hat{\mathbf{x}}_t^{[k]}, \Sigma_{\mathbf{x}_t}^{[k]}] = \text{PoseEst}(\mathbf{m}_{t-1}^{[k]}, \mathcal{H}_t, \mathbf{z}_t, \mathbf{x}_t^{[k]})$;
        sample $\mathbf{x}_t^{[k]} \sim \mathcal{N}(\hat{\mathbf{x}}_t^{[k]}, \Sigma_{\mathbf{x}_t}^{[k]})$;
        for $i = 1$ **to** $\text{length}(\mathbf{z}_t)$ **do**
            if $\mathcal{H}_t(i) = 0$ **then**
                // New feature
                $\tilde{\mathbf{m}}_t^{[k]} = \text{InvMeasModel}(\mathbf{z}_t^i, \mathbf{x}_t^{[k]})$;
                add $\tilde{\mathbf{m}}_t^{[k]}$ to $\mathbf{m}_t^{[k]}$;
            else
                // Update feature
                $\mathbf{m}_t = \text{EKF}(\mathbf{m}_{t-1}, \mathbf{z}_t^i, \mathbf{x}_t^{[k]})$;
            end
        end
        remove unstable features of $\mathbf{m}_t^{[k]}$;
    else $w_t^{[k]} = w_{t-1}^{[k]}$; $\mathbf{m}_t^{[k]} = \mathbf{m}_{t-1}^{[k]}$;
    add $\langle \mathbf{x}_t^{[k]}, \mathbf{m}_t^{[k]}, w_t^{[k]} \rangle$ to $\bar{\mathcal{X}}_t$;
end
if $\mathbf{z}_t$ has been sampled **then**
    normalize $w_t^{[k]}$ of $\bar{\mathcal{X}}_t$;
    $\mathcal{X}_t = \text{Resampler}(\bar{\mathcal{X}}_t)$;
else $\mathcal{X}_t = \bar{\mathcal{X}}_t$;
return $\mathcal{X}_t$, $\mathcal{U}_t$ and j;

V. EXPERIMENTAL RESULTS

Experimental tests have been performed on the TGR Explorer powered wheelchair in an indoor environment. This vehicle has two driving wheels. The odometric system consists of two incremental encoders connected to independent passive

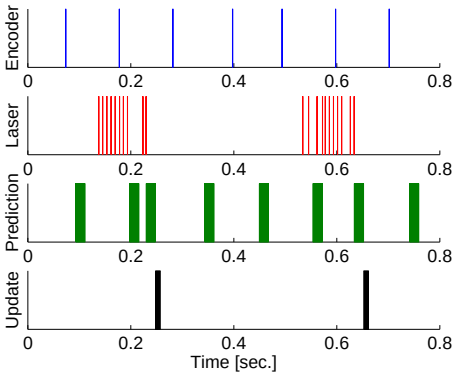


Fig. 4. Task Chronogram Example.

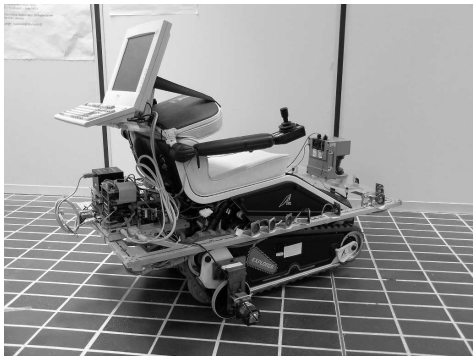


Fig. 5. TGR Explorer with data acquisition system for FOG sensor, incremental encoders and laser scanner.

wheels aligned with the axes of the driving wheels. A fiber optic gyroscope (FOG) HITACHI mod. HOFG-1 have also been used to accurately measure the angular velocity. The laser scanner measurements have been acquired by the SICK LMS200 installed on the vehicle. A resolution of 0.5° and a spectrum of 180° have been chosen in order to have a number of measurements that could simultaneously guarantee good map building and real-time implementation. The TGR Explorer powered wheelchair with data acquisition system is shown in Figure 5.

Different tests have been performed at the DIIGA Department for analyzing the performance of the proposed MR-FastSLAM algorithm. In this section, two significant tests are introduced and discussed: a loop-shape and L-shape experiments. Figure 6 shows the DIIGA Department taken from a CAD plan as well as pose estimations of both experiments with acquired with classicla MCL and assuming the map known. The purpose of the first experiment is to investigate if the multi-rate sampling aims to solve the loop-closing problem. The starting and ending pose of this experiment is $x = 38.4$m., $y = 21.8$m. and $\theta = \pi$rd. In the second experiment of a L-shape corridor, doors $\#1$ and $\#2$ where closed. The main difficulty lies on the U-turn done in the middle of the experiment where the robot has the greatest difficulty in estimating the orientation due to the lack of

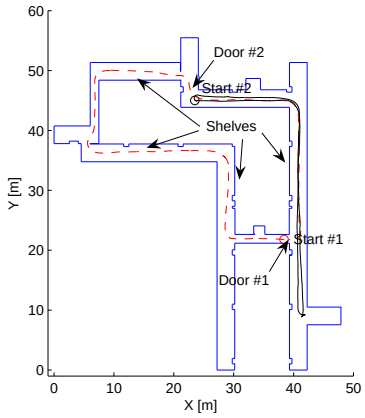


Fig. 6. CAD plane with loop-shape (dashed line) and L-shape (continuous line) experiments pose estimations with MCL.

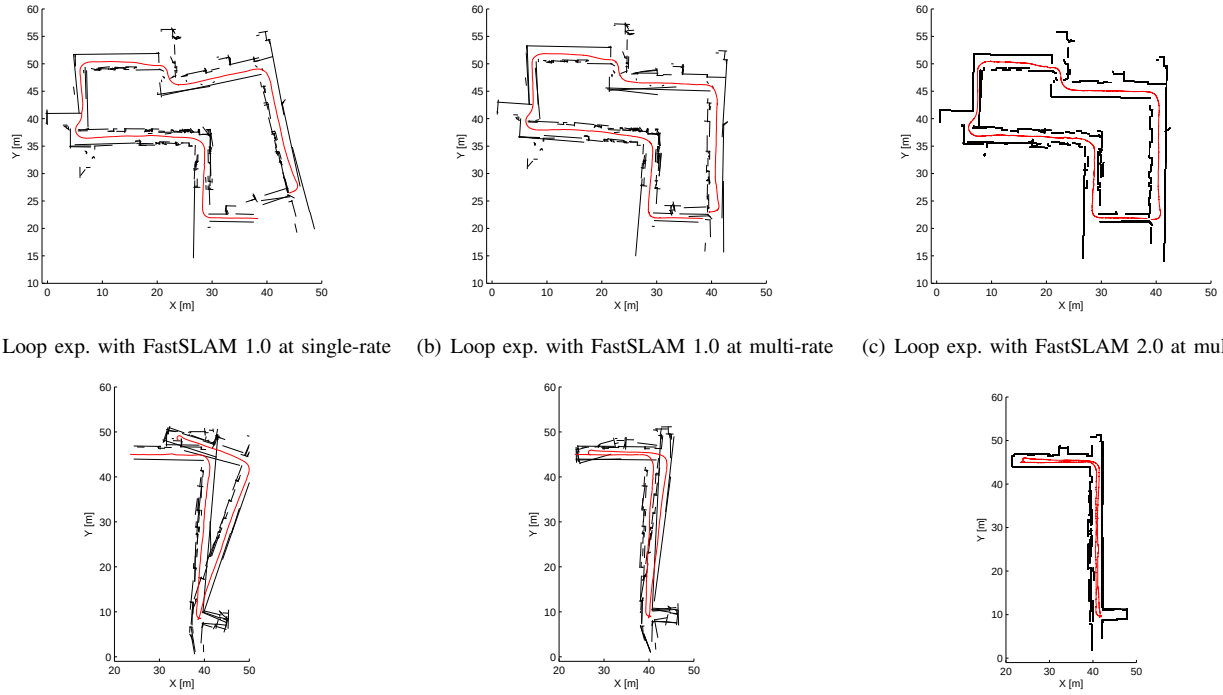
visible walls and the rate of turn. The starting and ending position of this experiment is $x = 23.5$m., $y = 45$m. and $\theta = 0$rd.

On the one hand, Figures 7(a) and 7(d) show the results obtained when using the single-rate FastSLAM 1.0 for both experiments (running at the laser sampling frequency), where clearly it fails the orientation estimation when turning. In this case, the number of samples used for the FastSLAM estimation is 100, probably not enough and therefore results may also be improved by increasing the number of particles, associated to a higher computational cost. On the other hand, Figures 7(b) and 7(e) show the results for the multi-rate FastSLAM 1.0, where it can be seen that it closes the loop and also improves the results of the the corridor experiment, but still has some difficulties in estimation the orientation.

In addition to this, the map building and also pose estimation robustness can be significantly improved when using the MR-FastSLAM 2.0, instead of MR-FastSLAM 1.0. Figures 7(c) and 7(f) show estimation results for the MR-FastSLAM 2.0 case, where it can be appreciated that the map estimation is much more accurate (corners with forms of 90°). In this sense, we can appreciate from Figure 8 that MR-FastSLAM 2.0 uses less number of features, which is also an indicator of its good accuracy, decreasing the computational cost, but without a detriment on pose and map estimation. The same performance has been also obtained in the other developed experiments which are characterized by simple paths along the corridors and laboratories of the DIIGA Department. Moreover, no specific assumptions have been considered on the structure and on the parameters of the considered mobile base. Therefore, the same performance of the proposed MR-FastSLAM algorithm can be obtained on different mobile bases with different kinematic models and parameters.

VI. Conclusions

In this paper, a new approach for multi-rate fusion, probabilistic self-localisation and map building for mobile robots have



(a) Loop exp. with FastSLAM 1.0 at single-rate (b) Loop exp. with FastSLAM 1.0 at multi-rate (c) Loop exp. with FastSLAM 2.0 at multi-rate

(d) Corridor exp. FastSLAM 1.0 at single-rate (e) Corridor exp. with FastSLAM 1.0 at multi-rate (f) Corridor exp. with FastSLAM 2.0 at multi-rate

Fig. 7. Experiment results.

been developed. In order to overcome the filtering problem of dynamic systems with inputs and outputs sampled at a different rates, asynchronous multi-rate holds have been used. In addition to this, an asynchronous fusion method (with asynchronous *prediction-update* steps) is also proposed. The prediction step is executed at least within a maximum period, which ensures a low discretization error and stability, while update step fuses measurements as they are received. Although in the paper this multi-rate asynchronous structure has been applied to PF in FastSLAM, it is clear that it can be used with many other filters such as EKF, UKF or Discrete-Bayes filters, since they also have the *prediction-update* steps.

Multi-rate asynchronous holds are hybrid systems that generate continuous signals from discrete sequences of inputs that may be uniformly distributed in time or not. The key idea is to use interpolation and approximation functions, or in general any primitive function, to extrapolate continuous signals. Thus, multi-rate holds act as interfaces for signals with different sampling rates, providing signals properly adapted to the required sampling period (discretization of the continuous signal). It this paper, it has been shown that general multi-rate holds reduce extrapolation errors in comparison to the naive solution of keeping the last updated value (ZOH). This is specially interesting in data-missing problems such as communication failures.

These ideas have been successfully applied to the multi-rate fusion of laser ranger and odometry measurements, by considering each sensor at their own sampling frequency rate. In this sense, the laser ranger is sampled at a slow

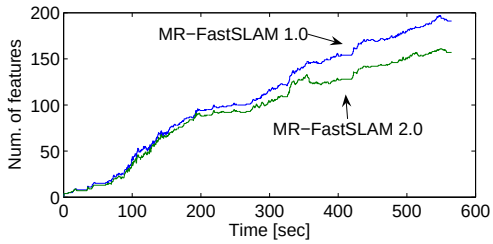


Fig. 8. Number of estimated features for MR-FastSLAM 1.0 and 2.0.

sampling rate while the odometry is sampled several times faster. The ideas of this paper can be also extended to other sensor fusion applications, such as vision and inertial fusion. In fact, in [31] and [33] preliminary ideas of multi-rate (synchronous) EKF and UKF were successfully applied to the mobile robot localisation problem (laser and encoder fusion) and 3D tracking problem (vision and inertial fusion).

As a result, the paper shows that a significant improvement can be obtained on the localisation and map building problem when the proposed multi-rate filtering approach is applied. Results show that multi-rate fusion improves the estimation with respect to the single-rate one for the same number of particles, since the multi-rate filter has lower discretization errors. Alternatively, similar errors are obtained with less number of particles. Therefore, reducing the number of particles reduces the computational cost.

The paper also provides a LS pose estimation method based on detected lines in indoor environments. It has been shown that the method is robust and accurate enough with a lower computational cost than Araujo's method [24]. The LS

method can successfully be applied to the FastSLAM 2.0 as demonstrated through experimental results.

Finally, the paper describes a novel object detection method, which is mainly based on multiple line fitting of landmarks (walls) with regular constrained angles. This method is particularly indicated for indoor structured environments and represents an improvement with respect to the standard LS line fitting method without significant incrementing the computational cost.

To conclude, methods for object detection, LS pose estimation and asynchronous multi-rate filtering are combined to produce a robust and efficient overall method for mobile robot localisation. These methods have been validated with experimental real data, in mobile robot moving on an unknown environment for solving the SLAM problem.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Government (MCyT), research project BIA2005-09377-C03-02 and by the Italian Government (MIUR) research project PRIN 2005097207.

REFERENCES

[1] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] S. Julier and J. Uhlmann, "Reduced sigma points filters for the propagation of means and covariances through nonlinear transformations," in *American Control Conference*, vol. 2, 2002, pp. 887–892.

[3] N. Gordon, D. Salmond, and A. Smith, "Novel aproach to nonlinear/non-gaussian bayesian state estimation," *IEE-Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.

[4] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Trans. on Signal Proc.*, vol. 49, no. 3, pp. 613–624, 2001.

[5] A. Smith and E. Gelfand, "Bayesian statistics without tears: A sampling-resampling perspective," *Americal Statistician*, no. 2, pp. 84–88, 1992.

[6] J. Carpenter, P. Clifford, and P. Fernhead, "An improved particle filter for non-linear problems," Department of Mathematics, Imperial Colege, Tech. Rep., 1997.

[7] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *UAI*, 2000.

[8] P. Jensfelt and H. Christensen, "Pose tracking using laser scanning and minimalistic environmental models," *Trans. Robotics and Automation*, vol. 17, no. 2, pp. 138–147, 2001.

[9] G. Zunino and H. Christensen, "Simultaneous localization and mapping in domestic environments," in *Int. Conference on Multisensor Fusion and Integration For Intelligent Systems*, 2001.

[10] J. Castellanos, J. Neira, and J. Tardos, "Multisensor fusion for simultaneous localization and map building," *Trans. on Robotics and Automation*, vol. 17, no. 6, 2001.

[11] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization," *Mach. Learning Autonomous Robots*, pp. 29–53, 1998.

[12] F. Dellaert, D. Fox, W. Burgard, and T. S., "Monte carlo localization for mobile robots," in *Int. Conf. on Robotics and Automation*, 1999, pp. 1322–1328.

[13] A. Bonci, G. Ippoliti, L. Jetto, T. Leo, and S. Longhi, "Methods and algorithms for sensor data fusion aimed at improving the autonomy of a mobile robot," in *Advances in Control of Articulated and Mobile Robots*. Berlin, Heidelberg, Germany: STAR (Springer Tracts in Advanced Robotics), Springer-Verlag, 2004, vol. 10, pp. 191–222.

[14] A. Bonci, G. Ippoliti, A. L. Manna, S. Longhi, and L. Sartini, "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*, December 2005.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Slam: A factored solution to the simultaneous localization and mapping problem," in *Proc. of the AAAI*, 2002.

[16] P. Colaneri and G. de Nicolao, "Multirate LQG control of continuous-time stochastic systems," *Automatica*, vol. 31, pp. 591–596, 1995.

[17] J. Tornero, R. Piza, P. Albertos, and J. Salt, "Multirate LQG controller applied to self-location and path tracking in mobile robots," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001, pp. 625–630.

[18] D. Lee and M. Tomizuka, "Multirate optimal state estimation with sensor fusion," in *American Control Conference*, 2003, pp. 2887–2892.

[19] P. Khargonekar, K. Poolla, and A. Tannenbaum, "Robust control of linear time-invariant plants using periodic compensation," *IEEE Transactions on Automatic Control*, vol. AC-30, pp. 1088–1985, 1985.

[20] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that probably converges," in *Proc. of the IJCAI*, 2003.

[21] G. Bourhis, O. Horn, O. Habert, and A. Pruski, "An autonomous vehicle for people with motor disabilities," *IEEE Robotics and Automation Magazine*, vol. 7, no. 1, pp. 20–28, 2001.

[22] S. Fioretti, T. Leo, and S. Longhi, "A navigation system for increasing the autonomy and the security of powered wheelchairs," *IEEE Transactions on rehabilitation engineering*, vol. 8, no. 4, pp. 490–498, 2000.

[23] E. Prassler, J. Scholz, and P. Fiorini, "A robotic wheelchair for crowded public environments," *IEEE Robotics and Automation Magazine*, vol. 7, no. 1, pp. 38–45, 2001.

[24] G. Araujo and M. Aldon, "Line extraction in 2d range images for mobile robotics," *Journal of Intelligent and Robotic Systems*, no. 40, pp. 267–297, 2004.

[25] R. Duda, P. Hart, and D. Stork, *Pattern Classification, Second Edition*. New York: John Wiley and Sons, 2001.

[26] G. Araujo and M. Aldon, "Optimal mobile robot pose estimation using geometrical maps," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 1, pp. 87–94, 2002.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[28] A. Doucet, "On sequential simulation-based methods for bayesian filtering," CUED/F-INFENG/TR.310, University of Cambridge, Department of Engineering, Tech. Rep., 1998.

[29] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods In Practice*. Springer, 2001.

[30] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," CUED/F-INFENG/TR 380, Cambridge University Engineering Department, Tech. Rep., 2000.

[31] L. Armesto and J. Tornero, "Slam based on kalman filter for multi-rate fusion of laser and encoder measurements," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2004, pp. 1860–1865.

[32] A. Huster and S. Rock, "Relative position sensing by fusing monocular vision and inertial rate sensors," in *Int. Conf. on Advanced Robotics*, 2003, pp. 1562–1567.

[33] L. Armesto, S. Chroust, M. Vincze, and J. Tornero, "Multi-rate fusion with vision and inertial sensors," in *Int. Conf. on Robotics and Automation*, 2004, pp. 193–199.

[34] J. Tornero, Y. Gu, and M. Tomizuka, "Analysis of multi-rate discrete equivalent of continuous controller," in *American Control Conference*, 1999, pp. 2759–2763.

[35] J. Tornero and M. Tomizuka, "Dual-rate high order hold equivalent controllers," in *American Control Conference*, 2000, pp. 175–179.

[36] ——, "Modeling, analysis and design tools for dual-rate systems," in *American Control Conference*, 2002, pp. 4116–4121.

[37] L. Armesto and J. Tornero, "Dual-rate high order holds based on primitive functions," in *American Control Conf.*, 2003, pp. 1140–1145.